$2.00

kilobaud T.M.

The Small Computer Magazine

ISSUE # 5 May 1977

Articles

Features

Old Fashion VALUE in the U.S.A. Tradition

Our 6800 computer system represents the best value available today, with no sacrifice in performance.

I would like to explain why this is true. The most basic reason is that the 6800 is a simpler, more elegant machine. The 6800 architecture is memory oriented rather than bus oriented as are the older 8008, 8080 and Z-80 type processors. This is an important difference. It results in a computer that is far easier to program on the more basic machine language and assembly language levels. It also results in a far simpler bus structure. The 6800 uses the SS-50 bus which has only half the connections needed in the old S-100 (IMSAI/MITS) bus system. If you don't think this makes a difference, take a look at the mother boards used in both systems—compare them. The SS-50 system has wide, low impedance 0.1 lines with good heavy, easily replaced Molex connectors. The S-100 bus, on the other hand, has a very fine hair-like lines that must be small enough to pass between pins on a 100 contact edge connector. I'll give you one guess which is the most reliable and noise free. As for cost—well any of you who have purchased extra connectors for your S-100 machines know what kind of money this can run into. The 6800 is supplied with all mother board connectors. No extras, or options like memory, or connectors for the mother board are needed in our 6800 system.

The 6800 is not beautiful, but "Oh Boy" is it functional. That plain black box is strong and it has an annodized finish. This is the hardest, toughest finish you can put on aluminum. Most others use paint, or other less expensive finishes. The 6800 does not have a pretty front panel with lights and multicolor switches. This is because the lights and switches are not only expensive, and unnecessary, but also a great big pain to use. We don't crank up the 6800; we use an electric starter—a monitor ROM called Mikbug. He automatically does all the loading for you without any time wasting switch flopping. So in the 6800 system you don't buy something expensive (the console) that you will probably want to stop using as soon as you can get your hands on a PROM board and a good monitor.

That's another thing. Mikbug® is a standard Motorola part. It is used in many systems and supported by the Motorola software library in addition to our own extensive collection of programs. It is not an orphan like many monitor systems that are unique to the manufacturer using them and which can only run software provided by that manufacturer. Check the program articles in Byte, Interface and Kilobaud. You will find that almost all 6800 programs are written for systems using a Mikbug® monitor. Guess how useful these are if you have some off-brand monitor in your computer.

The 6800 will never win any beauty prizes. It is like the Model "T" and the DC-3 not pretty, but beautiful in function. It is simple, easy to use and maintain and does its job in the most reliable and economical way possible. What more could you want?

Mikbug ® is a registered trademark of Motorola Inc.

SwTPC 6800

Computer System

with serial interface and 4,096 words of memory $395.00

☐ **Enclosed is $395 for my SwTPC Computer Kit** ☐ **Send Data**

☐ **or BAC**________________ **#**__________

☐ **or MC**________________ **Ex Date**__________

NAME

ADDRESS

CITY STATE ZIP

Southwest Technical Products Corp., Box 32040, San Antonio, Texas 78284

Q: HOW CAN YOU MAKE YOUR Z80 SYSTEM RIVAL THE EXPENSIVE PROFESSIONAL MINI-SYSTEMS?

A: BY ORDERING TDL's ADVANCED INTEGRATED SYSTEM SOFTWARE. TODAY.

ALL TDL SOFTWARE IS RELOCATABLE.

THE ZAPPLE MONITOR

The versatile Zapple Monitor maximizes the user's system executive control. All I/O drivers for all TDL software are contained in the Monitor; once the Monitor is modified to your I/O configuration all other TDL software is immediately compatible with your system making your total system setup extremely simple.

The Monitor supports a CRT, teletype and audio cassette as well as providing exceptionally powerful hardware and software debugging capability. It's modular organization also allows other user provided I/O drivers to be 'tacked on' at the end to handle any form of I/O your system may demand. The Monitor's 27 commands occupy only 2K of core. A description of the command set requires more space than available here.

Coupled with TDL's other software, your system attains the highest level system integration available for a micro-processor. $25

THE RELOCATING MACRO ASSEMBLER

Here is the most sophisticated programming tool yet developed for a micro-processor. It is without equal in terms of number of functions, scope of capabilities and usefulness.

- It generates a fully relocatable object code.
- It has complete macro generation and infinite nesting of macros capability.

TDL's ingenious Z80 opcode set has as a subset the same opcodes as the 8080. Where other Z80 functions resemble 8080 functions, the TDL mnemonics are logically derived for ease of learning. Where our Z80 opcodes have no 8080 parallel, we generally use Zilog opcodes. Your current 8080 source can be reassembled with only nominal text editing. $50

ZAPPLE BASIC

The most complete and innovative 8K Basic interpreter written. Highly compatible with other Basics. Unique Features include:

TRACE: Allows switching among various line numbers during execution.

SWITCH: Allows switching among various I/O devices either in the program or from the keyboard.

RENUMBER: Allows upward or downward renumbering of the numbered line statements.

LVAR: Lists program variables and their values on your command.

LLVAR: Same as LVAR, but outputs the information to your hardcopy device.

EDIT: A unique feature in an 8K Basic. Allows you to change the internal structure of a line with the 'delete' command.

ZAPPLE BASIC is unique versatile and powerful and up to 20% faster than comparable Basics. $50

TEXT OUTPUT PROCESSOR

TDL's general purpose word processor for the Z80. Used in conjunction with the TEXT EDITOR and MONITOR, it occupies 3K of core and provides powerful word-processing capability.

TDL's TOP features automatic paging, concatenation and justification, as well as many other formatting functions.

Straightforward entry of the Processor's numerous commands directly into the text puts you in total control. The output from the Editor is then run through the Processor to produce output exactly as you have commanded.

Imagination alone limits the possible applications of this word-processor. $50

SUPER BASIC

Our powerful new 12K Basic Interpreter has all the features of Zapple 8K Basic, plus these additions, and many others:

- Allows eleven digit precision including all built-in functions as well as SIN and ARCTAN.
- Multi-line recursive user defined functions.
- Sophisticated editing and program loading.
- Statements may be tabbed horizontally and/or spread over successive lines.
- AUTO inserts line numbers automatically into your program.
- A PRINT USING statement is included which allows the format to be specified as to either a string or a statement number.
- EXCHANGE interchanges the values of two variables at high speed.
- LOADGO allows one program to call in and to execute another program.
- CALL allows you to call resident assembly language subroutines with any number of numeric arguments.

SUPER-BASIC does much more and qualifies as the finest Basic available for microprocessors. $95

THE ZAPPLE TEXT EDITOR

This omnipotent tool gives you the ability to create or to change assembly language programs and to facilitate general word processing.

Text lines and characters may be located, inserted, deleted or changed at will in either forward or backward directions via a movable internal pointer.

Control may be effected by 24 alpha numeric commands which may be strung together in macro-like statements yielding superlative editing capability. $35

PACKAGE A - THE BASIC & SUPERBASIC PACKAGE: Consists of the Zapple Monitor, Zapple Text Editor, Macro Assembler, The Text Output Processor and either 8K Basic @ $ 169 or Super Basic @ $ 199

PACKAGE B - THE WORD-PROCESSING PACKAGE: Consists of the Zapple Monitor, the Text Output Processor and the Zapple Text Editor .... $ 85

PACKAGE C - THE SOFTWARE DEVELOPMENT PACKAGE: Consists of the Zapple Monitor, Zapple Text Editor and the Relocating Macro-Assembler .... $ 85

All TDL Software is supplied on paper tape in relocatable hex format along with TDL's comprehensive User's Guides and Manual. Write for prices on disk or cassette media.

ORDERING INFORMATION: Send check, money order or BankAmericard, Master Charge current number and expiration date. Shipment is usually made via UPS or UPS Blue Label. Specify other arrangements if you wish. Prepaid orders are shipped postpaid.

Contact us for hardware systems as well. Distribution Rights Available.

TECHNICAL DESIGN LABS

RESEARCH PARK BLDG. H 1101 STATE ROAD
PRINCETON, NEW JERSEY 08540 (609) 921-0321

I Print "Publisher's Remarks" TO No End Run

Wayne Green

Atlanta June 18-19th

The first big computerfest for the southeastern computerists is being held at the downtown Atlanta Marriott hotel – coordinated by the Atlanta Area Microcomputer Hobbyist Club. There are over 100 booths and the crowd expected is large. This is in conjunction with the Atlanta Hamfestival, which last year brought in over 5,000 . . . with the computer exhibits the biggest pullers of all.

The Marriott has 1000 rooms set aside for the show with special computerfest rates of $18 single and $24 double. Call their toll free number for reservations (800-228-9290).

In addition to quite a few talks by manufacturers about their new equipment, there will be quite a few ham oriented computer projects such as a computer controlled repeater, two computers conversing in Morse code, a system that will speak Baudot to a Teletype or to another system, and a demonstration where a computer will display any Morse code sent by a ham on a key.

Don Alexander is expected to be there with a computer controlled repeater . . . and his computer RTTY setup . . . plus most of the major microcomputer system manufacturers.

Ed Roberts, the president of MITS, is scheduled to speak . . . a first for a computerfest. Ed is the chap who got all this started . . . just a little over two years ago.

Atlanta is a place for the whole family to have fun. Just out of town is Stone Mountain, which has all sorts of entertainment . . . and there is Underground Atlanta, which really comes alive at night . . . bluegrass music, a bunch of fine restaurants, games (including those computerized games) . . . some people will drive all the way to Atlanta just to have a dinner at Aunt Fanny's.

I'll be there . . . around the 73-Kilobaud booth . . . giving some talks . . . and anxious to say hello to you and get any ideas . . . answer questions . . . help spark articles, etc.

There will be a whole raft of prizes including a door prize of a computer . . . and a brand new car. Atlanta is going all out this year to put on the biggest and best show in that part of the country . . . watch out Dayton!

Avuncular Advice

Since only about 10% of the computer hobbyists have a system up and running, it stands to reason that around 90% of us are looking for some advice and encouraging words from the pioneers. How about lending a helping hand . . . just because you had the guts to go out and blow $2000 on a piece of equipment, hoping you'd be able to get it working, is no reason why you can't pass along words of encouragement to those less foolhardy or adventurous.

For instance, if a good friend of yours asked you what kind of system you'd recommend for someone with a top of $1000 to spend . . . what would you suggest? How about a lid of $1500? . . . or $2000? I know that just about every reader will be interested in hearing from those who have been into the woods, whether they survived or not.

How about a letter for the *Kilobaud* letters pages with your ideas on the best system you think a hobbyist can put together for the above three (or any one of them) price categories? Double space your typing please . . . and no Teletype stuff, our typesetter goes berserk when she is handed all capital material. You might be able to help a hobbyist save a lot of money by giving him the benefit of your experience.

Bionic Boom

An interesting question was raised in *Newsweek* recently . . . as more and more home computers are set up, will we be able to tap into data banks the way larger computers can? For instance, will we be able to access our records for Blue Cross? IRS records? Police records? Many of these are interconnected now and recent laws protect our right to know what is on file about us.

Will the time come when we can write our own plane tickets? Make our own hotel and airline reservations? Will the IRS set up an interactive program which will help small businesses and the general public do their own tax forms?

It is fun to be plugged into "today" with a digital wristwatch, an HT on our belt, a CB in our car . . . and our own computer at home. Heh, heh . . . eat your hearts out, general public.

Buyer Beware!

One of our readers sent in an ad clipped from another magazine for a computer training course being sold by one of the mail-order schools. He reports that the course was very expensive, of low quality, and the "computer" involved was a bummer. He advises that this is *not* a good way to learn about computers. The computer furnished seems to come with 16 bytes of RAM, however a memory expansion kit is available which will double this to 32 bytes (not kilobytes).

I've had some correspondence about this correspondence school before . . . an ex-employee of the school wrote to tell me how crummy the school was.

It might be prudent to save your money for computer gear instead of a mail-order course on computers. If I get a chance I'll try to get some mail-order courses and review them . . . if any seem worth their salt, you'll read about them in *Kilobaud*. I wonder if any readers have come across a worthwhile mail-order computer course? Please advise.

Since many veterans can get the GI bill to pay the freight for study courses, it would seem that the first school to come up with some good equipment and a hot course might do well. How'd you like to end up with a Z-80 system and 64K of memory, floppies, etc . . . all paid by good old Uncle?

No Spika Da Ingles

One of the drawbacks to publishing program listings in KB is that they are written for one particular chip . . . say a 6800. This reduces the value somewhat to 8080 owners. We do try to get authors to provide the flowcharts, if they've used 'em, as an aid, but a better answer would be translations into the more popular instruction sets for machine language programs.

Should you get fun from this sort of exercise, this is a solicitation for machine language translations into 6800/8080/6502/Z-80 instructions of any machine language articles run in KB.

Conventions

Philadelphia, May 28-29. Computerfest. Contact Personal Computing, Conference & Exposition Management Co., Box 844, Greenwich CT 06830.

Cleveland, June 10-12. Computerfest.

Trenton, April 31-May 1. Computerfest. Contact Jaci DiPaolo, (609) 771-2487.

continued on page 21

20 Print "Editor's Remarks" 30 End Run

John Craig

Further Reflections on Computer Clubs

Last month I used these pages to get up on my soapbox and share some thoughts and observations concerning computer clubs. I mentioned that this interest in club activities had been brought about due to a newcomer in our local group expressing dissatisfaction with the way our club was being run. I went to a lot of trouble to take a survey of the membership and then discuss the results at the following meeting. Guess who wasn't there? Right, the newcomer who prompted the whole thing! (Incidentally, the name of our local group was officially changed from "The Central California Computer User's Group" to "The Micro-8 Computer Club." We're not affiliated with the now-defunct *Micro-8 Newsletter,* which was somewhat of a pioneering effort in this hobbyist movement, but we're certainly going to keep the name alive with our group ... since it all started at the school where we hold our meetings.)

You know, that incident with the new guy not showing up at the meeting is typical of another characteristic found in any club or organization. That is, that there are *always* going to be just a handful of "doers" in a club ... with the majority just going along and with the idea of getting what they can out of the organization. I think it's unreasonable to expect anything else. The phenomenon is based on human nature and nobody should get upset about it. This is one of the nice features of our "nonclub" approach. We don't have any activities to speak of. Therefore, there aren't any burdens placed upon the members ... and as a result, we don't have even a hint of conflict between those who do everything and those who do nothing. Interestingly enough, in the survey I conducted there was an overwhelming majority in favor of having speakers at some of our meetings. But, when it came time to decide who was going to take the bull by the horns and be responsible for getting a speaker lined up occasionally *not one person volunteered.* No big deal, though. We're going to be just fine ... with or without speakers.

I had some comments last month on the quality of speakers who get up in front of computer clubs and I'd like to devote just a little space this time to the person who is responsible for conducting the meeting, the club president. I believe that he can contribute more than anyone else to making each meeting a success ... and if he's turning people off then he should certainly be replaced. I recently attended a meeting of a club down in Southern California in which the person conducting the meeting straddled a turned-around chair for the length of the entire meeting! There were several occasions during the course of that meeting when I had the strongest urge to ask him if he was doing okay with keeping his blood pressure under control with all the excitement he was generating! It was ridiculous! He set the tone for the whole meeting ... and if I was a newcomer I doubt if I would have been back for more. And, the idea of newcomers getting turned off is the thing that bothers me the most.

Speaking of newcomers, there certainly should be efforts toward making them feel at home and letting them know there are a lot of people within the group who are willing to help out. I would think some of the most fruitful club projects would be those directed toward the new members. Remember how thirsty you were for knowledge when you first started?

If you've got a system up and running why don't you stand up at the next meeting and invite some of those newcomers to stop by your place some evening for a little get-together? You'll enjoy showing off your system ... they'll enjoy seeing it ... you'll both enjoy the exchange of questions and answers ... and a great time will be had by all (I hope).

Then we have newsletters. I get newsletters from clubs all over the country. Whether it's a large newsletter or just a two or three page flyer they all seem to have one thing in common. They're *all* constantly urging the members to contribute material! And, if they don't have any original material from members, the pages are filled with reprints of material from other sources. It seems, in most cases, that a single-page flyer sent out monthly (or even occasionally) would be adequate for getting significant messages out to the members.

Don't get me wrong when it comes to newsletters, though. I enjoy seeing articles written by the members. As a matter of fact, I've contacted the authors of several articles I've seen in newsletters and asked them to write for *Kilobaud*. I can also understand and appreciate that the newsletters provide a means for the local computer stores to get the word out to their customers.

Heathkits

Someone recently told me that he knew of several people who were waiting for the Heath microcomputer kit and that they weren't going to make the plunge until it came out. This surprised me because I hadn't heard of it before. Although,

continued on page 22

kilobaud T.M.

PUBLISHER
Wayne Green
EDITOR
John Craig
MANAGING EDITOR
Kurt Schmidt
ASSISTANT EDITOR
Jody Wright
EDITORIAL ASSISTANT
Peggy Sysyn
MANAGER
Biff Mahoney
PRODUCTION DEPARTMENT
Manager:
Lynn Panciera-Fraser
Staff:
Craig Brown
Gayle Cabana
Robert Drew
Michael Murphy
Bob Sawyer
Noel R. Self
Robin M. Sloan
TYPESETTING
Barbara J. Latti
Sandy White
Marie Walz
PHOTOGRAPHY
Bill Heydolph
Tedd Cluff
DRAFTING
Bill Morello
Lynn Malo
ASSOCIATE EDITORS
Don Alexander
Tim Barry
Sheila Clarke
Rich Force
John Molnar
Peter Stark
COMPTROLLER
Knud E. M. Keller
ASSISTANT COMPTROLLERS
Marge Nielsen
Robin Hunter
MARKETING
Sherry Smythe
Karen McDonough
Lisa Joseph
ADVERTISING
Bill Edwards
Gary Dozier
Leslie Bailey
Nancy Cluff
Barbara Hann
Janet Ames
Lisa Healey
CIRCULATION
Dorothy Gibson
Nancy Chandler
Janette Dyer
Florence Goldman
Judy Main
Jeane Shattuck
Theresa Toussaint
PURCHASING
Susan Brumaghim
COMPUTER DATA CONTROL
Judy Waterman
Judy Brumaghim
Sherry Dean
Mary Jo Sponseller
COMPUTER ENGINEERING
C. Robert Leach
David E. Wilensky
Richard Dykema
PRINTING
Michael Potter
John Bianchi
William Cering
Brent Lawler
Gary Steinbach
INVENTORY CONTROL
Marshall Raymond
Duane Morgan
Gary Slamin
PLANT MAINTENANCE
Bill Barry
Lorraine Pickering

Kilobaud is published monthly by 1001001, Inc., Peterborough NH 03458. Subscription rates in the U.S. and Canada are $15 for one year and $36 for three years. Outside the U.S. and Canada, please write for foreign rates. Application to mail at second class postage rate pending at Peterborough NH 03458 and additional mailing offices. Phone: 603-924-3873.

INCLUDE OLD ADDRESS AND ZIP CODE WITH ADDRESS CHANGE NOTIFICATION.

HARDWARE & SOFTWARE REVIEWS

EDIT, Version 2.0
Text Editor
Lorin S. Mohler
LSM Engineering
PO Box 3243
Orange CA 92665
Magnetic Tape Cassette or Paper Tape Plus Documentation, $22.50

A fairly good summary of what to expect from a text editor is given by the author of EDIT in the introductory section of his manual. "The text editor (EDIT) is a program that allows the user to create or modify ASCII files such as source files coded in FORTRAN, BASIC or assembly language. It allows editing on character, string, line or page levels. At any of these levels the user may make additions, insertions, replacements and deletions of the text."

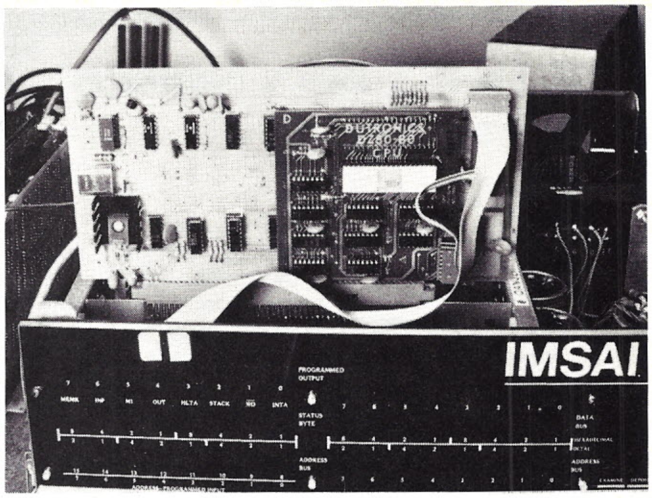
EDIT performs all these functions in an efficient manner and, once mastered, is relatively easy to use. The text can be either program/data, as implied in the author's introduction, or pure prose. In either case EDIT is a powerful tool, equally useful to the professional programmer or the computer hobbyist. (Note: Tarbell magnetic tape format for the 8080. – Ed.)

The "single character" command allows the user to enter a character that does not appear on his keyboard or one that requires special handling. The user supplies a decimal integer from 1 to 127 followed by the character I. The decimal integer is the numeric code for the desired ASCII character. This code is then interpreted as the desired character in all subsequent operations.

EDIT also provides limited macro capability. The user can define and retain a command string. This command string can be executed either once or repeatedly by issuing a single command. Unfortunately, only one such macro can be defined and retained. Defining a new macro automatically destroys any existing macro. The capability for multiple macros is an area the author may wish to explore in subsequent versions.

EDIT also provides unusual input/output capability. Input/output routines for paper tape, magnetic cassette and console device (TTY or equivalent) are part of the editor itself, eliminating the need for the user to create his own I/O routines.

EDIT is, in fact, an easy to use program. Regrettably, this fact is not immediately obvious from the manual. The documentation is written for the experienced practicing programmer. For a reader of that level it is excellent documentation, probably better than the industry average. For the hobbyist with limited programming experience, however, it is far less satisfactory. To paraphase an old line, it tells you how the watch is built – not how to tell time.



The DZ80-80 shown installed on the CPU board of an Imsai 8080.

For many hobbyists the documentation will take some deciphering. Those who make the effort will be rewarded with a useful tool. A summary of EDIT commands is given in Table 1. This is intended only to convey the scope of the program and no detailed analysis of the commands will be attempted. However, some abbreviations and conventions should be explained.

1. CP is the abbreviation for Character Pointer.

2. Control characters entered by the user are denoted by a superscripted "c" and the character that would have been printed if the CTRL key had not been depressed (i.e., P^c).

3. The ESCape key, entered by the user, is echoed by the processor and represented in the examples as the $ symbol.

4. When a string parameter is part of a command it is represented in italics.

Art McDonough
Ted Lincoln
Santa Ana CA

Z-80 Plug-in board
Upgrade regular 8080 CPU to Z-80
Dutronics, PO Box 9160
Stockton CA 95208
$159.95

There have been several reasons why I haven't gotten around to upgrading to a Z-80 until now and two of the most prominent were the *cost*, and the fact I would have an *extra CPU board* laying around afterwards. I recently ran across a new product from Dutronics (in Stockton, California) which solved both of those problems. As a result, I now have a Z-80 based Imsai up and running.

Command	Function
Z	ABORT – present operation and return to control of the monitor.
D	DUMP MEMORY – locations A1* through A2* to the command device.
E	EXCUTE – starting at memory location A1.
H	HEX ARITHMETIC
IB	INPUT BINARY FILE – starting at memory location A1.
IL	INPUT LOADER FILE (also known as HEX or INTEL format file)
M	MODIFY MEMORY – location A1.
OB	OUTPUT BINARY FORMAT – to File I/O from memory (A1 through A2).
OL	OUTPUT LOADER FORMAT – to File I/O device from memory locations A1 through 2.
R	RELOCATE MEMORY LOCATIONS A1 through A2 to a new area and starting address.
S	SEARCH MEMORY – starting at location A1 for character "H."
T	TOP OF STACK – returns the top of stack to your display device.
V	VERIFY – memory locations A1 through A2 for hardware errors.
	*A1 = First Address, A2 = Last Address

Table 1. Command Summary.

Table 1. Summary of commands available within the Dutronics monitor system.

Command	Special Characters (immediate execution)
RUBOUT	Deletes and echoes the last character entered.
C^C	Cancels a current command string or halts its execution.
I^C	Inserts a tab character within any string.
T^C	Turns off tab simulation if on; turns it on if off.
	Opening a Peripheral
<*hex address$*	Opens the user peripheral input.
>*hex address$*	Opens the user peripheral output.
	Input From User Peripheral Routine
Y	Clears the previous contents of the edit buffer and reads the next page.
A	Reads the next page and appends the input to the current contents.
	Output To User Peripheral Routine
P	Writes the entire edit buffer with a final form feed.
*n*P	Writes *n* lines from the CP and a final form feed.
PW	Writes the entire edit buffer without a final form feed.
*n*PW	Writes *n* lines from the CP and no final form feed.
	End File
E	Performs end.
	Buffer Contents
=	Returns the total number of lines/characters in the edit buffer.
@	Returns the number of the line in which the CP resides.
T	Types out all the text in the buffer (to the console).
*n*T	Types out *n* lines starting with the current CP.
	CP Control
B	Moves the CP to the beginning of the edit buffer.
Z	Moves the CP to the end of the edit buffer.
L	Moves the CP to the beginning of the current line.
±*n*L	Moves the CP forward (+) or backward (-) *n* lines.
*n*J	Moves the CP to the beginning of the *n*th line.
±*n*M	Moves the CP forward (+) or backward (-) *n* characters.
S*string*$	Moves the CP to the first character after *string.*
	Addition/Insertion
I*string*$	Inserts a string of characters at the CP.
*n*I	Inserts a decimal value of a single character at the CP.
I^C*string*$	Inserts a tab character as the first character of a string of characters.
	Deletion/Substitution
K	Deletes the entire line, no matter where the CP is located on it.
*n*K	Deletes from the current CP forward over *n* carriage returns.
±*n*D	Deletes forward (+) or backward (-) *n* characters.
C$string_1$\$$string_2$$	Changes $string_1$ to $string_2$.
C*string*$$	Deletes *string* (changes it to null).
	Macros
XM *command string* $$	Defines a macro command string.
*n*X	Executes the macro *n* times.
XD	Deletes the macro.
	Exiting Edit
G *hex address*$	Goes to an external user routine.
	Search Commands (Repeats from above)
S*string*$	Moves the CP to the first character after *string.*
C$string_1$\$$string_2$$$	Changes $string_1$ to $string_2$.
C*string*$$	Deletes *string* (changes it to null).

Dutronics has come out with a "piggy-back" Z-80 board which is mounted on a 4" square PC board. It is called the DZ80-80 and is designed to simply plug into the 8080 socket on your present CPU board. The real advantage, besides the price, is that it will fit Altair, Imsai or just about any 8080-based system. (I was also able to incorporate the board into my Polymorphics system, using the instructions supplied by Dutronics.)

Installation

To say the least, the installation is *simple.* The time involved amounted to approximately 20 minutes, most of which was spent reading over the manual. The photo shows the board installed on the Imsai CPU board. The installation involved removing the 8212 Status Latch located at the right of the DZ80 board so the header plug at the end of the ribbon cable from the board could be plugged in there (to replace the 8212). The only other step involved removing the 8080 chip itself and plugging the new board into the empty 40-pin socket. Needless to say, one of the advantages of going the DZ80-80 route is that you can very easily go back to your original 8080 CPU if you desire.

Software Compatibility

One of the considerations in moving up to a Z-80 is the fact that all the 8080 software will also run on it. All of the 8080 software that I've written for my Imsai ran without a hitch. And, speaking of hitches, it seems there is an exception with regard to *all* 8080 software running on a Z-80 machine. Some sophisticated programmers have been known to store information in the parity flag. Because of this I've heard that versions of MITS BASIC (below 4.0) will not

continued on page 21

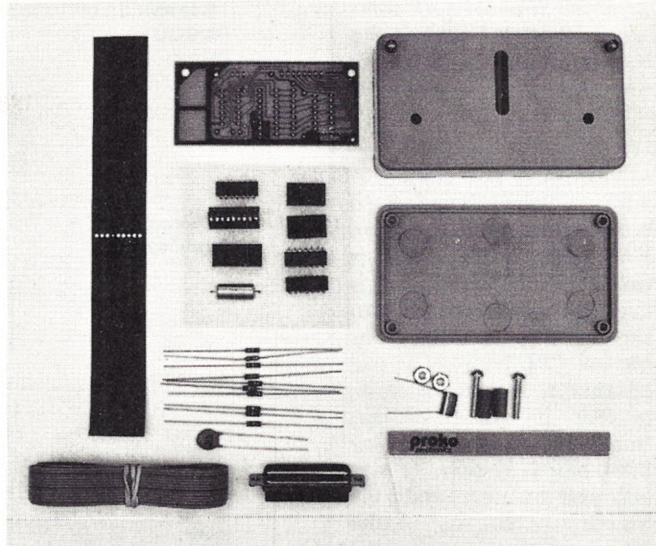
NEWS OF THE INDUSTRY

LOW-COST PAPER TAPE READER

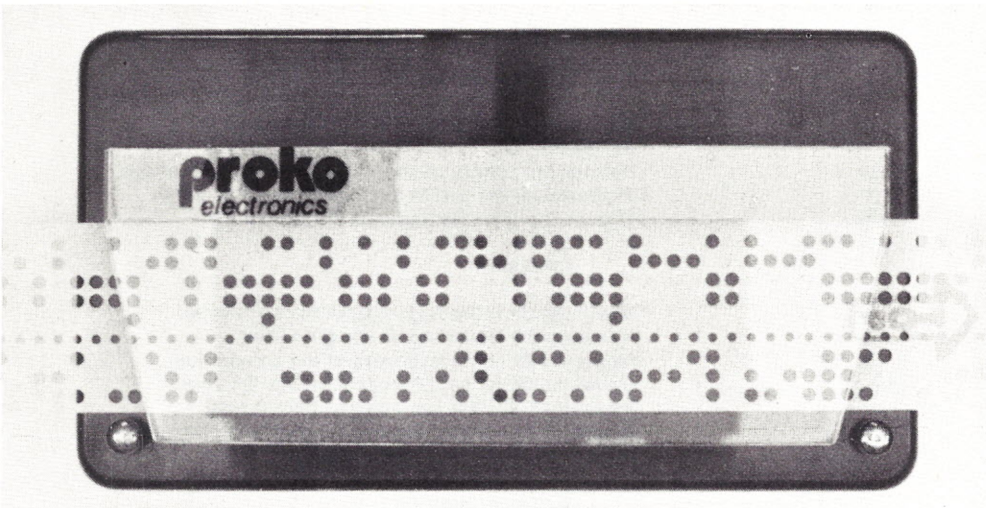
Why would you buy a PROKO paper tape reader if you owned a TTY with a paper tape reader? First is to save your hearing and nerves from permanent damage by listening to the clinkity-clank of that noisy reader. Second is also to help save your nerves by preventing you from getting a bad case of impatientitis (a common affliction among computer hobbyists working with 10 cps equipment).

Why would you buy a PROKO paper tape reader if you don't have paper tape input capability with your system?" Simple. So you would have the capability! There's just too much hobbyist software running around in paper tape to ignore the fact you need this capability.

The PROKO paper tape reader is fully self-contained (i.e., the components are not exposed to the outside world), is easily assembled, has no moving parts, and loads paper tape programs faster than a speeding bullet (or is that "more powerful than a locomotive"?). Price is $55.00 in kit form and $68.00 assembled. For further information contact PROKO Electronics, 439 Marsh St., San Luis Obispo CA 93401.



Low-cost Paper Tape Reader from PROKO (kit).



Low-cost Paper Tape Reader from PROKO (assembled).

ASDC INTRODUCES ALTAIR BUSINESS SYSTEM

The Altair Software Distribution Company is pleased to announce the introduction of a comprehensive set of software packages designed for the small business system market. The Altair Business System includes complete software packages for accounting, word processing and inventory management. The software may be licensed for use in individual packages to accommodate the needs of retail stores, small wholesale distribution centers, industrial users, professional firms and other business offices.

The new Altair Business System is designed around the Altair 8800 Computer. The system hardware may be individually configured for each installation and typically includes a CRT terminal, a typewriter-quality precision printer and one or more floppy diskettes.

The Altair Software Distribution Company (ASDC) is a subsidiary of MITS, Inc., the company that pioneered the use of microcomputers by the general public in 1974. Since that time, more than 8000 Altair Computers have been purchased for commercial, industrial, governmental, educational and personal use. MITS recently established the ASDC for the purpose of providing continuing software support to the users of Altair Computers. The ASDC seeks to acquire or develop quality software packages for distribution through the more than 25 Altair Computer Centers throughout the country.

The Altair Business System software is packaged in modules to allow a purchaser to select the components of a system that will most closely fit his needs. The accounting package is comprised of four modules – general ledger, receivables, payables and payroll. The word processing package is a flexible text editor system that allows large volume text material, such as contracts or other lengthy documents, to be stored, easily edited and printed. The inventory management package is a flexible data base management system which allows a business to keep complete inventory records "on line."

The component packages of the Altair Business System are available under a one-time fee licensing arrangement. The licensing includes three years of software maintenance. Each package is accompanied by a comprehensive set of documentation including operator guides and systems guides as well as training aids.

The Altair Business System may be seen at local Altair Computer Centers located in major cities throughout the country. Additional information may be obtained at local Altair Computer dealers, or contact the Altair Software Distribution Company, Suite 343, 3330 Peachtree Road, N.E., Atlanta GA 30326. Phone: (404) 231-2308.

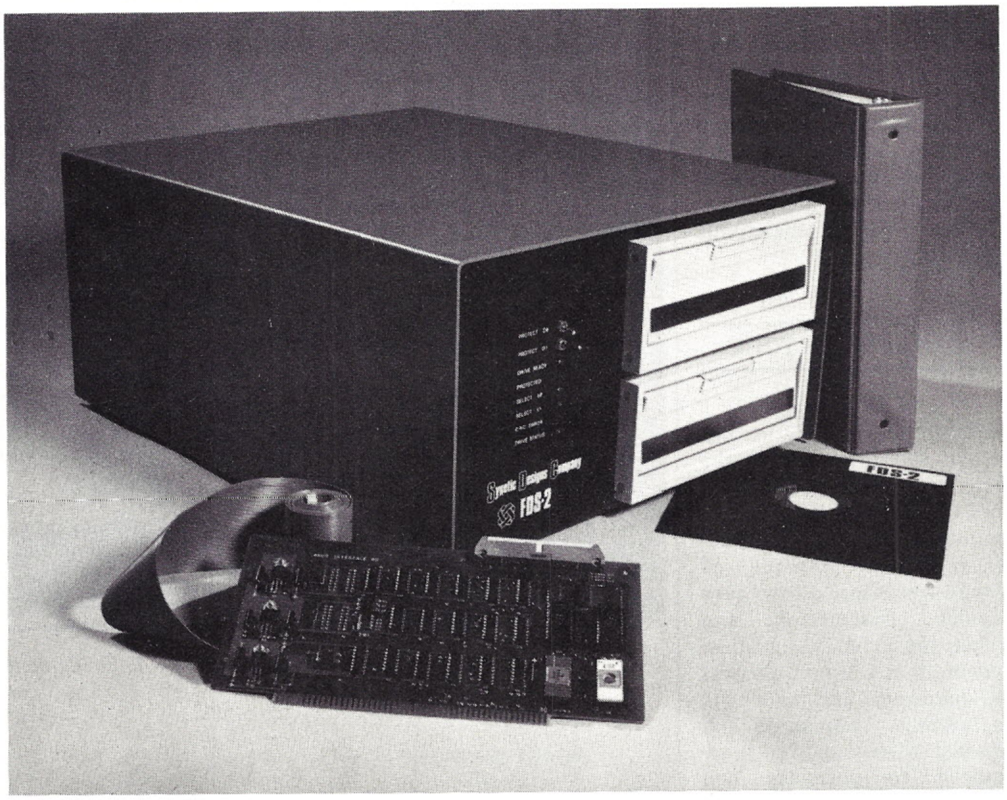
ALTAIR 7000 GRAPHICS/PRINTER

The new MITS Altair 7000 Graphics/Printer now makes electrostatic printing a fast, economical and extremely flexible process. The multifunction 7000 is a printer, plotter and graphics hard-copy output device that is plug compatible with the 680 and 8800 mainframes via one PIO port.

The flexibility of the Altair 7000 Graphics/Printer is due to eight print electrodes, driven directly by software, instead of the usual seven found in 5 x 7 matrix printers. Copies made from the printed output are actually more legible than copies of typed paper and can be made for about 1¢ per foot of electrosensitive paper.

When the 7000 is used as a line printer, characters are generated using a 5 x 7 dot matrix. Altair BASIC supports three different sizes of character sets (each with upper and lower case) to produce line widths of 20, 40 or 80 characters in the four-inch wide printing area. The speed is 160 characters per second (80 characters per line) or 120 lines per minute. Different character sizes are selected with the CHR$ function in BASIC.

The eighth or extra printing electrode in this unit provides symmetry along the horizontal and vertical axes to permit plotting. With the vertical distance between electrodes equal to the distance between lines, there's no gap from line to line.



Model FDS-2 floppy disk system from Synetic Designs.

This special feature makes the new 7000 ideal for graphics. Pictures can be produced that show either a distinct outline or a sophisticated, detailed picture with shaded areas. When the eight-dot columns are printed close together, the effect is a very dark image. When the columns are printed farther apart, the image appears lighter.

The 7000 is controlled by using a single port on an 88-4PIO parallel interface board. One section provides the eight bits of information to be printed and the other section provides control.

The new Altair 7000 Graphics/Printer will be available within 60 days of order placement at a cost of $785. Electrosensitive paper is also available through MITS.

For further information contact MITS, 2450 Alamo S.E., Albuquerque NM 87106.

READY TO USE FLOPPY DISK SYSTEM FOR S-100 8080 MICROPROCESSORS

Synetic Designs FDS-2 complete disk system includes dual floppy drives, controller, interface, power supplies, cabinet, and software. Utilizing iCOM's sturdy and proven IBM compatible Frugal Floppy™ system together with their excellent Executive system, Text Editor, and Assembler, the FDS-2 features a stylish cabinet and an exclusive "Executive Handler."

The system is delivered ready to use with no I/O vector assignments, initialization routines or program relocation required of the user. Simply insert the interface card into the computer, the customized diskette into the floppy drive, and RUN. A source copy of the FDS-2 Executive Handler is provided to allow the more sophisticated user to build a more complex disk operating environment.

Delivery is stock to two weeks with OEM and dealer discounts available. The ready to use disk system is distributed nationwide by the Byte Shops and other dealers. For further information contact Tom Kirven, Synetic Designs Company, POB 2627, Pomona CA 91766.

DIGITAL GROUP PHI-DECK INTERFACE TO ALTAIR BUS

The M712 from MicroLogic is an 8-bit parallel I/O port consisting of a bidirectional data bus and four hardware-generated strobe signals (2 input, 2 output). It is a single Altair bus compatible card and will operate with all Altair/Imsai/Sol/Polymorphic CPUs.

Although the M712 may be used as a general-purpose interface to any device which requires a bidirectional data bus it was specifically designed to interface the *Digital Group-PhiDeck* digital cassette system to the Altair bus. Interconnection is simple, only twelve signal lines are required from the Digital Group controller card, and MicroLogic supplies complete documentation and cabling.

Software is simplified since the M712 I/O strobe pulses are generated by hardware on the card, thus eliminating the need for clumsy software-generated strobe pulses.

Price is $69.95 kit, $79.95 assembled and tested. All MicroLogic PC boards are top quality, plated-thru, gold-plated contacts, and all ICs are socketed. For more information contact MicroLogic, PO Box 55484, Indianapolis IN 46220.

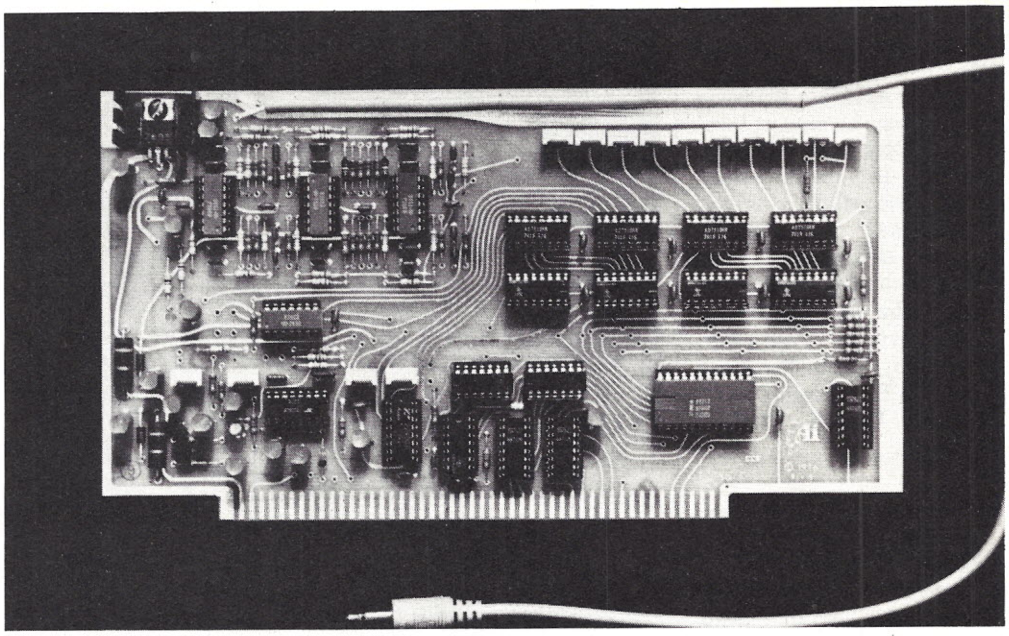
HOBBY-WRAP TOOL

If you have ever built a microprocessor kit, or any other complex digital device, you know soldering is not the ideal way to make hundreds of interconnections. The technique of "wire-wrapping" allows easy circuit construction, and when the inevitable mistake occurs it is a simple process to disconnect and rewire the offending connection. A special wire-wrap tool is required to correctly and securely "attach" the wire to IC sockets. These tools range from simple hand operated devices that are useful for very small jobs all the way to bulky AC operated devices with the associated power cord.

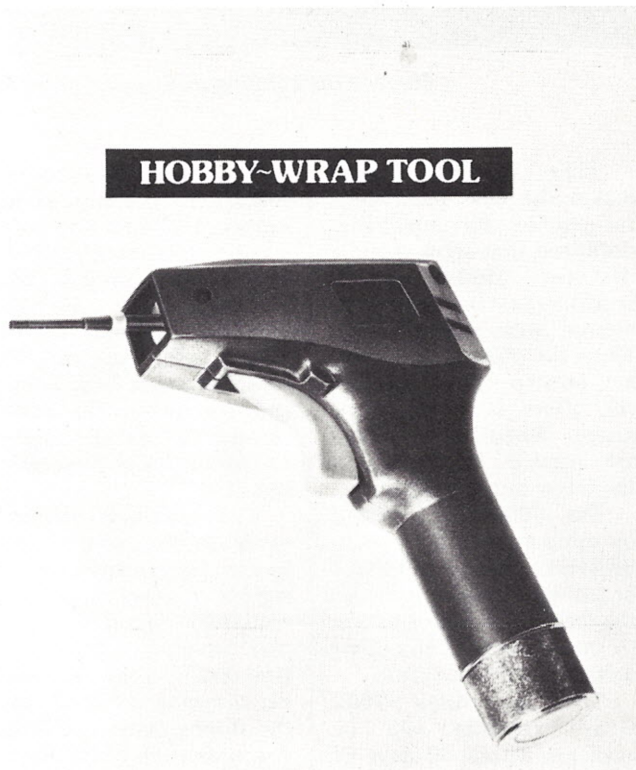
The best approach to wire-wrapping is the Hobby-Wrap Tool, manufactured by the OK Machine and Tool Corporation. This wrapping tool is motor driven for ease of use, yet is battery operated, eliminating the trailing power cord. The device is powered by two size "C" cells, and weighs only 11 ounces. The wire bit accepts standard 30 AWG wire for .025 inch square DIP stakes. The bit produces the "modified" wrap, which wraps a two turn layer of insulation around the stake for complete mechanical security. Each wrap is uniform, and the battery powered motor allows complete flexibility when maneuvering into tight corners on a CPU board.

The best feature of the rugged Hobby-Wrap tool, however, is the price ($34.95), which puts professional wire-wrapping within the means of every serious hobbyist.

For further information contact OK Machine and Tool Corporation, 3455 Conner St., Bronx NY 10475.



Speech Synthesizer from Ai Cybernetic Systems.



OK Machine and Tool's hobby-wrap tool.

THE TALKING COMPUTER MACHINE

The Model 1000 Speech Synthesizer is a hardwired analog of the human vocal tract. Various portions of the circuit simulate the vocal cords, the lungs, and the variable frequency resonant cavity of the mouth, tongue, lips, and teeth. All of the information necessary to produce the speech sounds of American English has been programmed into ROMs which reside on the synthesizer board. The unit accepts a string of ASCII characters (each character representing a particular phonetic sound or phoneme) in exactly the same fashion as a printing peripheral.

Because the synthesizer is primarily an analog circuit which is commanded digitally, new programming information is required only at the end of each completed phoneme. The maximum information transfer rate is very slow at about 50 bytes/sec (25 bytes/sec typical). The Model 1000 is directly compatible with the Altair/IMSAI bus structure. A demonstration cassette is available for $5. A programming manual which includes interfacing information costs $4. The price is $325.

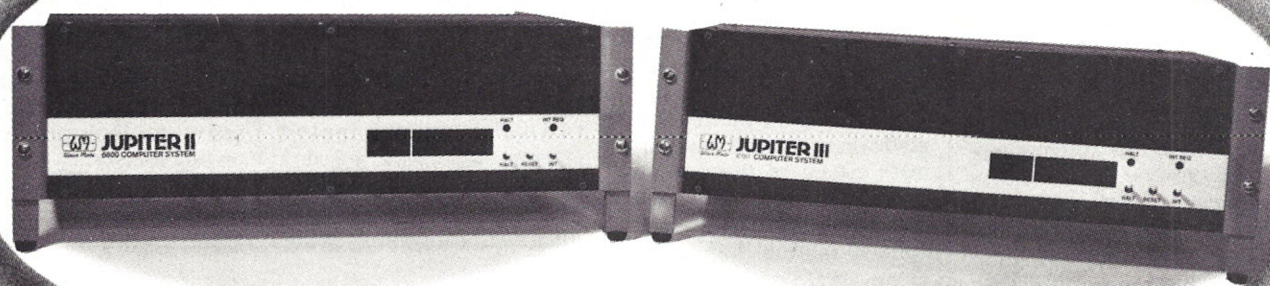
For further information contact Wirt Atmar, Ai Cybernetic Systems, PO Box 4691, University Park NM 88003.

HIGHER BAND RATE WITH PHASE ENCODING FOR "COMPUTER AID"

Computer Aid Model 3M3 Digital Data Recorder utilizes a radically new drive system for the 3M Data Cartridge (DC 300). This

continued on page 22

POPULARITY EXPLOSION!



JUPITER II A
6800 System
$795

ASSEMBLED

JUPITER III A
Z80 System
$865

ASSEMBLED

If you thought the quality of a wire-wrapped system was beyond your price range — Take a look at what we have now!

The Jupiter IIA and the Jupiter IIIA Basic computer systems. You get the system module cage with fully assembled backplane, fully assembled plug-in ferro-resonant power supply, front panel and your choice of 6800 or Z80 CPU module. All less than the price of the two best selling 8080 systems!

Plus you can choose from the fastest growing selection of memories and peripherals available from any manufacturer, like our 2KB EPROM/4KB RAM/serial RS-232 module and our new 1024 character video module. Both can transform your basic computer system into a real star.

And remember, all Wave Mate products meet the highest quality industrial standards, with rugged construction unmatched by anyone! Join the popularity explosion and get yours now! Write or call for more info and your closest Wave Mate authorized distributor.

You get your choice of microprocessors! And you get wire-wrapped modules too!

Now you have a low cost way to get started into personal computing without sacrificing future growth capability!

Send information on: ☐ Jupiter IIA system
☐ Jupiter IIIA system

NAME ______________________

ADDRESS ______________________

CITY ____________ STATE ________ ZIP ________

Wave Mate

WAVE MATE 1015 West 190th Street, Gardena, California 90248
Dept 24

Telephone (213) 329-8941

W-5

Look Ahead

Rich Didday

You may be wondering what Lookahead is all about, especially if you're a new subscriber to *Kilobaud*. Here's a restatement of the idea.

It seems that every time the forces of technology come up with a widely applicable new gadget, it sneaks into our society rather quietly at first. Four rough stages are identifiable. In the first phase, people who already have an interest in the gadget start using it and talking about it. In the second phase, people starting with no commercial or personal interest gradually learn of the gadget, become interested in it, and make the decision to indulge on the basis of fairly personal, close to home, factors. Will it make my life easier, more fun, richer? How much does it cost? Will my spouse agree to spend our money for it? What can I do with it? (The home/personal computer movement seems to be in the second phase right now.) For gadgets which gain mass acceptance, the second phase snowballs into a third phase in which so many people have the gadget that it becomes an accepted part of the *good life*, and people who otherwise might not even have dreamed of having the gadget get one simply because everyone else has one – it's the thing to do. Once the third stage is well established, and millions and millions of people own and use the gadget, the full effects of the gadget start to be felt throughout the whole society. In the mature, fourth phase, all the little side effects which were ignored, unnoticed, or ruled insignificant by early owners come to light – side effects which may well *be* insignificant for each individual, but which when multiplied by millions and millions of users make up a new set of problems and changes in our society as a whole.

The introduction, gradual acceptance, and ultimate effects of the automobile, television, the pill, disposable food and drink containers, the Xerox machine, etc., provide ample evidence of the process. Until rather recently, the side effects of technological advances have been ignored or at worst grouchily tolerated – taken as signs of *progress*. In a growth oriented society, in fact, the side effects of one gadget may be welcomed as providing openings for yet more gadgets which, in soothing the side effects of established gadgetry, cause yet more side effects, opening room for more gadgets, and so on in the infinite loop. In the last decade, however, it has become clear that we are starting to bump against the limits, that some of the flexibility is going out of the system. We no longer appear to have the natural resources, the energy (both physical and mental), or the space to withstand continuing waves of radical, unforeseen changes.

One solution we can rule out is that of simply stopping technological change. We needn't spend time trying to evaluate whether that would be a good thing to do – it simply seems to be impossible, given our current circumstances. Some people have tried, and failed miserably – one of the more colorful attempts was that of the Luddites. Ned Lud lived in Nottingham, England in the early 1800s. He and his fellow stocking weavers felt threatened by the introduction of mechanized looms. What makes us remember the Luddites was the way they expressed their concern – they stomped into the local mills and destroyed the looms. It's a sign of our society's attitudes about the inevitability of technological change that the Luddites are now thought of as buffoons.

If we as a society have neither the desire nor the ability to stop technological change, and we no longer have the ability to cope smoothly with rapid, radical, unanticipated change, what can we do? Perhaps we can develop ways to anticipate side effects of new technologies and try to gently guide their development away from potentially harmful outcomes.

Perhaps we can develop ways to anticipate side effects of new technologies and try to gently guide their development away from potentially harmful outcomes.

Well, that sounds nice, but how in the world can we do it? Maybe if the very people who are involved with a new gadget would take a little time to think about possible effects instead of concentrating solely on the commercial development and personal use of the gadget, maybe if the early users would make a little effort to observe the effects of the new gadget on themselves and their families, maybe if the first owners would make an effort to get people talking about the possibilities, maybe that would be an important first step. In other words, maybe the right approach is to develop a group consciousness, an awareness of the possible effects of mass acceptance of our gadget, the home computer. Lookahead is an attempt, an opportunity, to do that.

If we can bring it off, it will be novel in several ways. The only similar efforts I can think of all involve concerned scientists speaking out, giving warnings about obviously harmful effects of such developments as nuclear weapons, nuclear power plants, and certain lines of genetic research. Our gadget is a consumer item, is not controlled by any governmental or research organization and, most interestingly, doesn't have any obvious ill effects. But it most certainly will have large, pervasive psychological effects on our society.

Why is this forum called Lookahead? It's called Lookahead because just as the circuitry in a lookahead processor inspects upcoming instructions in the program and gets things ready for their execution, this forum provides all of us the opportunity to point out problems and developments that might otherwise elude our attention until they are upon us.

So what is Lookahead? It's not an attempt to predict the future, it's an attempt to get people talking about future possibilities, to share our observations, hopes, fears, and dreams. The hope is that by developing an awareness of the possible impact on our society, our society can be better prepared for the day when virtually every home has some sort of digital hardware, some sort of home computer/interactive TV in it.

Write:

Lookahead
1218 Broadway
Santa Cruz CA 95062

BOOKS

How To Get The Most Out Of Your Low-Cost Electronic Calculator
Ronald M. Benrey
Hayden Book Company, Inc., 50 Essex St. Rochelle Park NJ 07662
Publication 0-8104-5942-6
Paperback, 112 pages, $4.95

The first inclination for a reader of *Kilobaud* will be to dismiss this book as trivial. It deals with the basic operation of the simplest types of electronic calculators. But stop and consider the whole of the society rather than the very specialized segment represented by the computer hobbyist. The low-cost electronic calculator represents far more of a revolution than either the microcomputer or the $800 top-of-line calculator.

First of all, remember that even now, in 1977, most Americans have never operated any calculator of any kind. Of those who have, the vast majority have used only a basic 4-function machine which they obtained in the last two years. The low-cost machine is introducing automatic computation to housewives, to third graders, to retired longshoremen and to millions of people who have no previous experience with calculators and computers. When all of that is considered, this suddenly seems a much more useful and reasonable book.

The book starts with a brief and simplified introduction to the hardware and architecture of the calculator. This is followed by an explanation of the keyboard and of the basic arithmetic functions, including the use of a constant switch and a % key. Examples lead the reader step-by-step through sample problems of the most elementary level. These examples are very clear and easy to follow but are pitched to the level of a totally inexperienced student.

The remainder of the book is devoted to sample applications. The best of the applications sections is devoted to financial problems, such as mortgage payments, interest and annuity calculations, to name a few. The class of problem unquestionably represents the most common use of low-cost calculators. Every adult has some occasion to do computations of this type and these explanations are very well done, easily grasped by the least experienced reader.

The remaining applications sections are Household and Workshop, Hobbies, Mathematics and Fun Calculations. Most of these are of less general interest and some are trivial. They include Farenheit/Celsius conversions, mile/kilometer conversions, air-conditioning computations, wallpaper coverage, board-feet lumber calculations, and a variety of other more specialized applications. Many of these applications are of only transient interest in themselves. However, they will lead the novice to consider uses of the calculator other than balancing a checkbook or checking fourth grade homework.

The only quarrel with the book is the price. At $4.95 the cost of the instruction represents almost half the cost of the equipment. However, if a friend or relative of yours has acquired their first calculator you may wish to consider giving them this book. Or, if you're giving a low-cost calculator as a gift to a novice user, consider adding this book to the package. The total cost should still be under $20.00, a very inexpensive introduction to the endless possibilities of electronic computation.

A. H. McDonough
El Segundo CA

Microcomputer Primer
Mitchell Waite and Michael Pardee
Howard W. Sams & Co.
$7.95, 5"x8", 224 pages, paperback

This book is aimed at the person who has some background in digital electronics, little or no background in computing, no background in microcomputers, and who wants a guided tour of the main problems involved in building and using small microprocessor based systems. This goal is accomplished by providing a pleasantly written overview of the main ideas, coupled with a sampling of specific circuit designs from a variety of different systems, photographs of specific products, and an example of writing and hand assembling a small program.

Chapter 1 provides a brief perspective which outlines the history of microprocessors, and suggests possible applications. The book really starts with Chapter 2, which provides definitions of the basic constituents of any computer system (central processing unit, memory, I/O interfaces, I/O devices, software); mentions the concepts of registers, busses, interrupts; and displays photographs of some common I/O devices. Chapter 3 begins by giving a typical power supply schematic, a typical clock schematic, then turns to a survey of construction techniques (wire-wrap, vector board, card cages, etc.). Next, the book devotes 15 pages to a comparison of the electrical characteristics and architecture of the more popular microprocessors. After incorporating a nicely done overview of RAM and ROM memories, the chapter ends with a discussion of inexpensive I/O which includes circuit diagrams for and discussion of a simple toggle switch/LED front panel, an octal thumbwheel/seven-segment display front panel, a hex keyboard based front panel, an ASCII keyboard input device; and a brief mention of the idea of video displays. As throughout the book, the discussion is intended to give a person who can read logic schematics the basic idea of the problems involved without going into all the details necessary to incorporate the devices into a specific system.

Chapter 4 covers the basic ideas involved in writing machine language programs. It discusses several instruction types, explains a number of different addressing modes, introduces the notion of iteration, gives flowcharts for doing 4-bit unsigned binary multiplication and table lookup, and then, turning to specifics, goes through the design and writing of a block memory move program for a 6800 based system.

There are two appendices. The first covers the representation and manipulation of binary, octal, and hexadecimal numbers as well as mentioning three types of binary coded decimal representation. The authors have a nice idea for making complement arithmetic sound plausible – they ask you to think of a car odometer rolling backwards, which makes it seem reasonable to think of 99999 as representing -1, 99998 as -2, etc.

People interested in the design of memory chips will find the appendix on memories fascinating – it shows the basic circuits and describes the micro and macro characteristics of bipolar, MOS, PMOS, NMOS, CMOS, and SOS memories. If you've ever wondered why and in what way dynamic memory chips need to be refreshed, the answers to these and other chip design questions are here.

There are bound to be errors in the first edition of any book, especially one such as this which surveys a vast amount of topical, changing-even-as-you-write information. Since the authors' goal is to get the main ideas across, using specific examples only to highlight larger ideas, perhaps it is irrelevant that a number of specific details are stated fuzzily and, in some cases, erroneously. My hope is that by mentioning three such errors here I will give a clearer picture of the level the book is pitched at.

At the end of an interesting section which covers the use of ROM to replace circuit components (by storing the truth table of the components in the ROM and looking up output values instead of computing them), the authors include an example of using a ROM to implement a universal counter (three control lines select among binary count up, binary count down, BCD count up, BCD count down, etc.). A schematic and a truth table are shown. In the schematic, the four lines bringing the number to be counted up or down are drawn as if they were the three control lines. The truth table is (1) not a truth table and (2) lists the four data lines instead of the three control lines as it should. So, the advice is, if you're reading this book, and you come to a specific example you don't understand, don't agonize over it, just pay attention to the overall picture.

Another type of error creeps in when the authors base their generalizations on a limited number of cases. Most of the examples in the book seem to be drawn from the 6800 and the SC/MP chips. Thus, on page 147, we find the statement that in direct addressing, the most significant byte of the memory address appears as the second byte of the instruction, while the least significant byte appears as the third byte of the instruction. Well, that's certainly true for a lot of microprocessors, but the order is reversed in some, including the most popular of them all, the 8080.

The discussion of programming culminates in a program written for the 6800 which copies the contents of the second 256 bytes in memory into the corresponding positions in the third 256 bytes. Again, the discussion is reasonably clear and demonstrates the important points – but the program doesn't work as shown. The authors have misunderstood the basics of the instruction fetch cycle and have therefore consistently miscomputed the displacements in relative branch instructions. The program counter is incremented after each byte of an instruction is fetched, so after an instruction has been fetched, the program counter points to the location in memory one higher than the address of the last byte of the current instruction. The authors erroneously state (page 177) that the program counter points to the first byte of the current instruction, and this leads them to compute branch displacements which are too large by the length of the branch instruction, i.e., 2_{10}.

Overall, then, in *Microcomputer Primer* we have a book which does what it sets out to do. Errors in details are compensated for by the breadth of coverage, by the inclusion of a large amount of information on what goes on inside the chips, and by a pleasant style. It would make a good choice for someone with some electronics background who wants a broad introduction to microcomputers – for someone who isn't ready to care about the details just yet.

Rich Didday
Santa Cruz CA

Discovering Basic: A Problem Solving Approach
Robert E. Smith
Hayden Book Co., Inc.
224 pages, softbound, $6.85

There are two major requirements to using this text. The first is that you must have access to a BASIC terminal or a home system with BASIC in it. The second is that you must be willing to spend considerable effort to complete the learning process which is only started by most examples in the book.

This is definitely not a book for the beginner wishing to learn BASIC. The majority of the lessons in the book require the user to write programs from flowcharts. The book is divided into 41 lessons and 50 review problems.

The types of problems included in the book could be divided into the following categories:

Basic definitions & programming techniques	15%
Math routines	28%
Matrix routines	15%
Business & finance	6%
Games & puzzles	16%
Miscellaneous programs	13%
Tests and scoring	7%

I feel that too much emphasis has been placed on the definitions of the problems and too little emphasis has been placed on the programming techniques required to solve the problems.

The matrix programs use functions that are not available in most of our home BASIC systems. These include MAT, TRN, IDN, and INV.

I did find the author's approach to review tests to be interesting. A set of multiple choice questions is given. Then you are to enter your answers using DATA statements or INPUT statements into an evaluation program. Upon running the program, your test results are printed out.

The program listings, when included in the main part of the text, are printed in large type and are very easy to read. However, for most of the programs the reader must look in the section in the back of the book where the type is quite small and the programs are not always in an easy to find order.

In my opinion, you should "discover" BASIC with a more fundamental book and use the "problem solving approach" to sharpen the skills that you have acquired.

Gordon Flemming
Sylmar CA

CMOS Cookbook
Don Lancaster
Howard W. Sams Company, Inc.
4300 West 62nd St.
Indianapolis IN 46268
$9.95 Paperbound

Times change, especially in the world of digital electronics. It has only been a couple of years since the first surplus U1974 RTL chips were eagerly purchased by experimenters building their first digital circuits. The digital revolution had not reached the "home" enthusiast, and practical knowledge and application information was hard to come by. The only real source of information was a small paperback by Donald Lancaster called the *RTL Cookbook*. This reference provided practical knowledge concerning the techniques required to implement digital systems using RTL. However, RTL was quickly replaced by transistor-transistor logic (TTL), and soon the surplus market was flooded with a wide selection of the popular logic family. Most of the popular digital circuits built by today's experimenters are based on TTL. Lancaster again responded to the needs of the experimenter by providing the *TTL Cookbook*. Probably more home digital experimenters possess this book than any other reference, as it provides all the information required to understand and use TTL.

TTL has several shortcomings, however, especially in the area of power consumption. The newest

continued on page 23

Around the Industry

John Craig

The Intersil Intercept Jr. Tutorial System

I recently attended a one day course on the Intercept Jr. conducted by Intersil and their distributor, Schweber Electronics. The course is intended as an introduction to computers with an assumption being made from the start that the student hasn't any experience at all (i.e., just what I've been needing!). The class consisted of such people as buyers, salesmen, marketing types, electronic engineers, and others. As a matter of fact, before I was through I had a couple of them converted into hobbyists.

Obviously, the seminar wasn't intended for the hobbyist, or hobbyist applications, but I planned on attending with the viewpoint of the hobbyist in mind. In spite of this, I'm not going to discuss my impressions from the standpoint of a hobbyist. One of my reasons for not doing this is because Pete Stark has done such a fine job on that angle in his article which you'll find in this month's issue. I'd like to discuss the educational applications for the Tutorial System (and tutorial it is!).

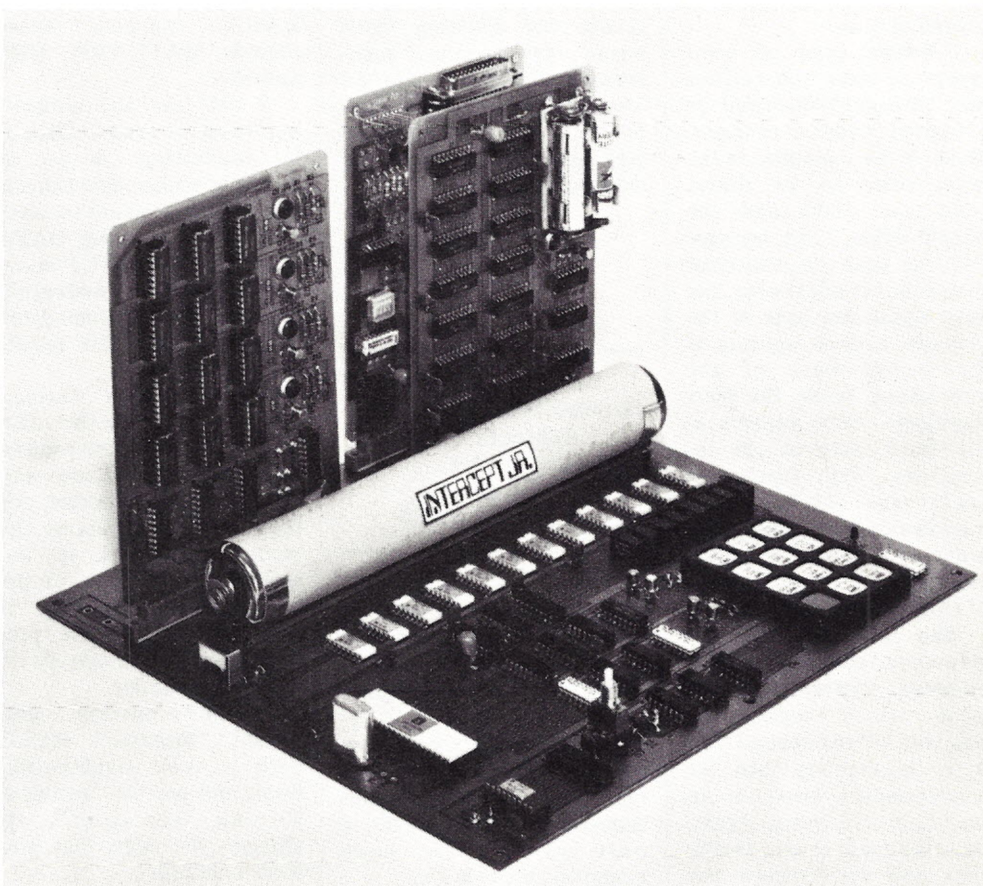
I was an instructor in computer systems for several years before I got into my present line of *work.* Because of this I find that whenever I sit in a class as a student I tend to be an *instructor observer* as well as a student. As a particular subject is presented I find myself reflecting on how I would have presented it. As I was playing that little game during the course it suddenly hit me as to what a fantastic educational tool the Intercept system really is. For a course in computer fundamentals I doubt if it could be beat . . . and I'll show you why.

Ah, but wait a minute. Somebody out there might be asking the question, "What does *Kilobaud* have to do with computers in Education?" My response would be that generally we don't address that subject and application unless it has to do with using the computer for educational purposes in the home. I feel that one of *our* objectives (all of us) should be to get as many people turned onto computers as possible. One of the ways we can accomplish this is to get the young introduced to them. Those of us in a position to do something in that area (and that includes teachers, members of the school board, the PTA, etc.) should be aware of the Intercept system (and others) so we can influence its use if, and when, the time ever comes to do so.

But, let me share with you some of the reasons why this thing turned me on so much. And by the way, I certainly don't think this thing is limited to secondary education . . . it could do the trick for teaching computer fundamentals just as well in colleges or industrial/commercial environments.

One of the most impressive features of the Jr. lies in the fact it doesn't require a power supply! It uses 4 D-size batteries which will last for several weeks due to the low current drain of the CMOS logic used throughout. This means that a course doesn't have to be taught in a lab or special classroom with a lot of electrical outlets . . . but even more important, all those power supplies aren't needed!

The Console Control keypad provides an input media for entering, examining and starting programs in memory. One of the nice features here is the fact the keys are coded with the PDP-8 instruction *mnemonics.* The student is actually doing machine-language programming . . . but using, and getting familiar with, the PDP-8 mnemonics. Needless to say, this will be invaluable for students who will eventually move up to programming an actual PDP-8. At the same time it will be providing an introduction to symbolic coding for all students, regardless of the machine they may encounter next. The ON/OFF switch does in fact remove power from the microprocessor and all the



Intersil's new Intercept Jr. low cost CMOS microprocessor tutorial system.

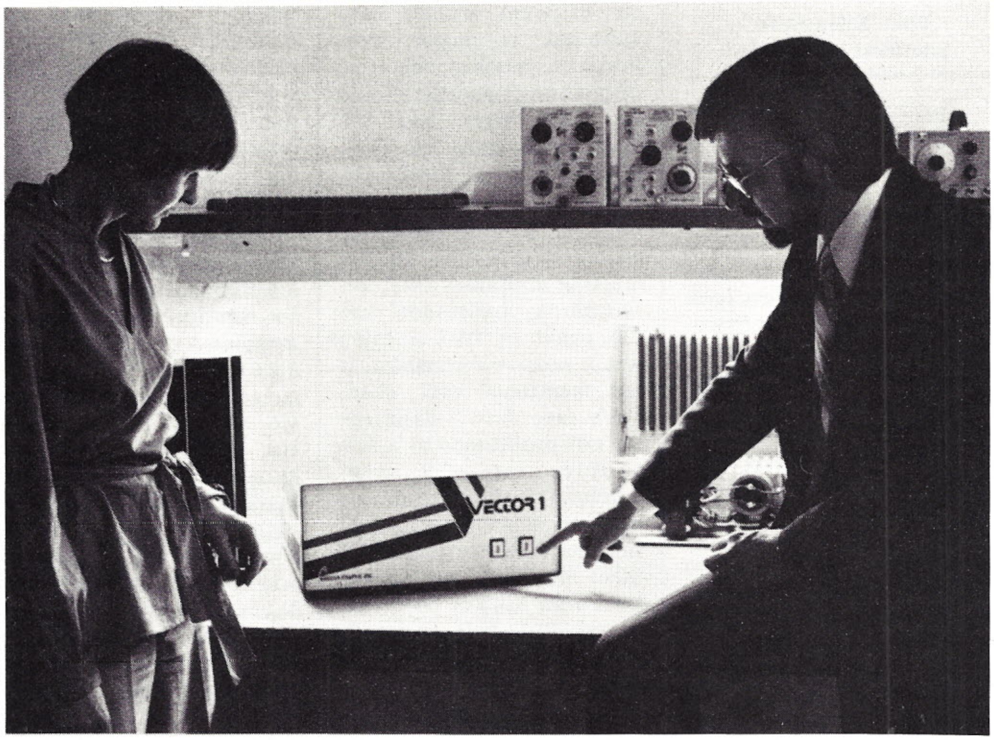
support chips . . . but it is also configured to provide for nonvolatile memory since power remains applied to the CMOS RAM chips.

The 6957 Audio Visual Module is the real *gotcha.* Within a matter of a few hours the student can be *hooked* on computers and computer programming by the fact he or she will be able to see *and hear* the results of a program which has been written, debugged and run. The seven-segment displays along with the binary displays and switch register can do a terrific job of serving as training aids for teaching all of the numbering systems used in computers. And, there is nothing like having the students throw the switches and actually generate the numbers being discussed.

A student could probably learn more in one day of using and programming this unit than he could in 30 days of lectures and studying computers from a textbook.

For further information contact:

Intersil
10900 N. Tantau Ave.
Cupertino CA 95014
or
Schweber Electronics
Jericho Turnpike
Westbury NY 11590



A deep technical discussion on the complexities of the Vector 1 front panel between Lore Harp and John Craig.

The Vector 1

Vector Graphic, Inc. is a somewhat unusual company in one respect, but for the most part it's doing the same thing a lot of other small companies around the country are doing these days. The *normal* thing they're involved in is designing and manufacturing a microcomputer system for the hobbyist and small business market. The *unusual* aspect is the fact the company is run by two women: Lore Harp and Carole Ely. I don't really consider that to be out of the ordinary so much and Lore and Carole don't either.

As you can see from the unposed photos included in this section the folks at Vector have managed to come up with a rather slick looking entry into the market. And, it's all theirs. They manufacture the rugged Vector 1 cabinet, the 18 slot mother board, power supply, 8080-based CPU board, 8K Static RAM board and a rather neat PROM/RAM board which we're going to be getting an article on in the near future. The list of commands provided in the PROM monitor program are shown in Table 1.

All of the subassemblies mentioned above are being sold separately or as a complete system in computer stores throughout the country. They're offering high quality products, and just as important, they've expressed genuine concern for providing good before and after-sales service to their customers. What more can I say? For further information contact:

Vector Graphic, Inc.
717 Lakefield Road, Suite F
Westlake Village CA 91361

Carole Ely and Lore Harp with their pride and joy, the Vector 1.

A	ASCII memory dump
D	HEX memory dump
G	Go to and execute program
L	Load program from Tarbell tape cassette interface and execute
P	Program memory from terminal
R	Record Tarbell tape
T	Test any block of memory, using a pseudorandom number sequence
V	Verify cassette tape

Table 1. The Vector 1 PROM/RAM board monitor commands.

The "Kill a Byte" Standard – Revisited

In the second issue of *Kilobaud* (February, 1977) we had an article by Hal Walker, President of National Multiplex, which was entitled "The 'Kill a Byte' Standard." In that article he more or less *took on* all of the competition, the various recording techniques, and the existing (and future) cassette recording standards. It was an interesting article and we've had some interesting feedback on it. Without a doubt some of the most interesting has been from the other manufacturers in the industry which Hal mentioned in his article.

We solicited comments from *all* the manufacturers he mentioned but a couple of them didn't reply. The ones who did had some very interesting comments to make regarding his article and the current state of affairs with recording of digital data on cassette. Here they are.

From Gary Kay, Southwest Technical Products . . . One of the reasons SWTPC is using the 300 baud AC-30 is because of the "Kansas City" standard. It just happened that we had not designed our system at the time of the symposium. As far as I am concerned, it is a little late to establish another audio standard. Audio tape is so inferior to the new mini-floppies that cassette standards will soon be a thing of the past. There is just no comparison between a disk and audio cassette computer system. Although the former will be more expensive. The system's power and flexibility is increased tremendously.

From Don Tarbell, Tarbell Electronics . . . This is in response to Hal Walker's article in the February issue of *Kilobaud*. I have so far avoided getting involved in the "standard" controversy. Generally, I feel that too much proclaiming of standards has happened already. At least six different articles have appeared in major publications, pushing one type of method or another. Some of these have had the audacity to state that their method *is* a standard, without even asking for further opinion. At least there was an open invitation to the meeting at Kansas City, although my impression that the result of that meeting was a *proposed* standard, is not mentioned very often. Let's face facts: standards are not proclaimed by a few people, especially manufacturers; they come to being over a long period of time because the *users* find them the best way to go. Even IBM has not been able to establish true standards in some areas (EBDIC vs ASCII).

However, in this case, it seems to me that some statements were made which, if not completely false, were at least misleading. First, I would like to make a note about BAUD rates, which can be misleading in themselves. The way that most people use the word, BAUD refers to the number of bits per second that is transferred from one medium to another. This doesn't sound like a problem at first, but when you consider that all of those bits are not always used to transfer *information*, it can become a bit confusing. For example, some people say that the BAUD rate of the Tarbell Cassette Interface is 1500, since that is the speed at which bits are transferred to and from the tape. When they then compare it to another interface that claims a rate of 1200, they think mine isn't much faster; or to another with a rate of 2400, they think is quite a bit faster. Let's say that at the 1200 BAUD rate, someone uses one start bit and two stop bits in addition to the eight data bits. That's eleven bits per byte, so the transfer rate of *information* is 1200 divided by eleven, or about 109 bytes per second, as opposed to 187 bytes per second with my method. Using this reasoning backwards, I claim that my equivalent "BAUD" rate is eleven times 187, or 2057 BAUD. My point is, if you want to compare transfer rates, ask how many bytes per second are transferred, not what the BAUD rate is.

Hal makes the comment that a start bit and stop bit are standard items, but fails to mention that that only applies to asyncronous systems, and not syncronous ones, like mine. He makes an objection to the checksum loader program, which I provide in the manual, but fails to mention that I also provide a very short bootstrap program, which can be as short as 17 bytes, depending on the version. The logic of his remark: "You can't have a standard unless everyone can use it" escapes me. I don't own an 8-level paper tape punch, therefore, I can't make standard paper tape. That's right! So what? If you want to read and write Tarbell formatted tapes buy an interface from me or from one of the other manufacturers that makes a compatible interface. With respect to the 800 bit per inch density, it's not part of my religion, or anything, I just thought it was a nice round figure, and it is still being used by most of the industry standard digital cassette units. I had used it in my hobby computer system since 1971, and I had a good feeling for it's reliability with regular audio cassette recorders – that's why I chose it. Incidentally, many people are using their Tarbell Cassette Interfaces at higher rates, with very good results. Many are using 2400 bits per second, for example, which is 300 bytes per second, and it only requires a 14k resistor across the 27k one to make the change! However, on the other end, some people with recorders that have poor frequency response, have reported having reliability problems even at 800 bpi. The above facts make me feel that I've come up with a reasonable tradeoff, so that on the one hand, a person doesn't have to buy an expensive recorder, but on the other hand, he has a reasonable transfer rate.

Now, one thing I'm *not* going to do is state that my format should be a standard. If people want to call it that, I'm certainly not going to argue with them! But I think more time is required to get used to the different types. My advice is: go over to a friend's house, or to a club meeting, and see what other people are happy with. My main efforts in the coming months will be to continue providing a high level of support for the Tarbell Cassette Interface, both in the way of repairs, improvements, and software.

From Harold A. Mauch, Percom Data Company . . . As a participant in the 'Kansas City' Symposium on cassette recording and manufacturer of several cassette interfaces for the Computer Hobbyists, I read with interest Hal Walker's article on cassette recording in *Kilobaud* #2. While in general I agree with his conclusions (FSK is inferior and a self-clocking technique is needed) I feel he overstated the case for 2400 Baud operation.

All of the PerCom cassette interfaces operate at 2400 Baud (PE 2400). However our experience is that the performance (reliability) at 2400 Baud is distinctly inferior to the performance at 1200 and 300 Baud (KC standard rate) *when used with ordinary audio cassette recorders.* We include the 2400 Baud rate in our cassette interfaces for those who wish to experiment or for those who use saturating mode recorders. I believe the tape equipment manufactured by Mr. Walker's company uses saturating mode and as such will work quite well with the PE 2400 waveform since the recovery amplifiers and circuits were designed for this waveform (page 129, Fig. 1). The situation is different with portable audio cassette tape units. The circuitry was designed for speech audio in which phase shift and group

continued on page 21

The BASIC Forum

Dick Whipple/John Arnold

In this month's BASIC Forum, we are pleased to present the first of what we hope to be a long series of responses from *Kilobaud* readers. We had no difficulty deciding on a topic since nearly all the letters dealt with the same subject; namely, ideas for new BASIC statements. If you recall in our first BASIC Forum, we asked reader opinion of the efforts to standardize the BASIC language. We further wanted to examine the effect such standardization would have on the creativity of the software oriented hobbyists who wanted to write their own interpreters. To date, we have not received any specific responses to the standardization question, but we have rattled the cages of several software types who have already begun using their creative skills thinking up new BASIC statements. We want to present their ideas (which is the ultimate purpose of BASIC Forum), but first we thought it would be of interest to readers to give a few observations on the proposed ANSI Standard for Minimal BASIC.

In 1974, the American National Standards Institute established a committee to develop a standard for a Minimal BASIC language. A report published in January, 1976, summarizes the standards voted on and accepted by the committee. We have just finished looking over the report and thought a brief statement of our impressions would be in order. First off, the standard definitely represents a minimal BASIC. Fig. 1 is a summary of these statements most of which are easily recognized even by the beginning BASIC programmer. The OPTION statement, however, is not well known and apparently represents a compromise among members of the committee. It is used to set the beginning array argument to either zero or one. Some BASICs like MITS start arrays at position zero while others, like older Data General BASIC, start theirs at one. The OPTION statement would permit improved compatibility between programs written on such differing systems. We have heard criticism of the inclusion of this statement since virtually none of the currently available BASICs have it. Those who have had experience modifying programs to run on different machines certainly recognize the array argument as a thorny problem. Perhaps the OPTION statement represents the only solution satisfactory to all concerned, but it does leave one wondering how many other thorny problems yet to come will be solved likewise. This brings us to another point. The committee is currently working on what they call enhancements (or what you might call extensions) to the Minimal BASIC standard. They certainly have their job cut out considering the current proliferation of extended BASICs available on the hobby/commercial market. Although threads of similarity run through them, everybody is vying for the title of *Innovator of the Year.* Innovation is generally looked on as healthy, but we wonder if such rapid changes in the language produce severe problems for a committee creating standards *ex post facto* as it were. In closing out this topic, we would like to add one last observation. The committee has attempted to establish an upward compatible standard; that is, a foundation on which more elaborate BASICs can be built. The same philosophy will probably prevail in the creation of enhancements to Minimal BASIC. It appears that they have not tried to establish standards that restrict language embellishments. For this they should be applauded. By the way, comments concerning ANSI BASIC can be directed to the American National Standards Institute, 1430 Broadway, New York NY 10018.

Now to our readers' suggestions. The first comes from David Price, 3901 Victoria Lane, Midlothian VA 23113. David proposes a new BASIC statement structured as follows:

```
ANSWER L1,L2
```

The ANSWER statement would prompt with a question mark "?" like an INPUT statement then await user entry of either a Yes CR or NO CR. A YES would cause branching to L1 while a NO would cause branching to L2. Actually, the interpreter would only check the first character of the replay (a "Y" or "N") before branching. An incorrect response would cause an error message to be printed. The situation requiring a "YES-NO" branch is quite common especially in game programs. David's example use of the statement is shown in Fig. 2. In addition, David has expressed an interest in collaborating with someone experienced in interpreter writing. Apparently, ANSWER is but one of his ideas for new statements.

Another letter comes from Erik Brown, 3655 W. 6th St., Winona MN 55987. Erik writes that he is a high school senior using MITS extended BASIC in his ALTAIR 8800. He passes along two ideas for new BASIC statements:

1. An optional parameter (or parameters) on RUN, LIST, PRINT, and INPUT statements to permit Input/output port modification. This would be useful for operating different I/O devices such as printers, plotters, and terminals (possibly remote) without having to generate entirely different BASIC routines for each. Creating and debugging programs could be speeded up by first selecting a fast CRT terminal and then later switching to a slower TTY terminal to get hard copy or to punch a paper tape. A new statement called INITIALIZE would have to be included to set up the type of I/O device for a particular port option. The INITIALIZE statement would permit specifying such information as device type (serial or parallel) and set up the proper masking of the control port.

2. His second suggestion is the creation of a new BASIC statement to handle interrupts. It would appear something like Example 1 for single level interrupts or Example 2 for vectored interrupts. Interrupts during program execution would cause branching to the indicated line. The vectored interrupt option of line 20 would also permit establishment of priority relationships. Two additional commands would also be needed:

```
ENABLE INTERRUPTS
```

and

```
DISABLE INTERRUPTS
```

From Erik's suggestion it seems clear that he is interested in a fairly sophisticated multi-I/O operation for his Altair and Basic. We

continued on page 23

```
10 ON INTERRUPT GOTO 100
```

Example 1.

```
20 ON INTERRUPT(5) GOTO 200
```

Example 2.

Letters to the Editor

UNANSWERED QUESTIONS

I enjoyed issue one of *Kilobaud*. A lot of information has been stuffed between the covers of the magazine! There is one thing that could be improved: the depth of the articles. I have been involved with computers for the past three years and I am beginning to appreciate the value of articles that cover many levels. All of the articles I read in *Kilobaud* covered important topics, but the author sometimes redundantly covered some areas while missing others.

One example was Chris Bowick's article on control of the outside world by computer. Although he gave an excellent description of the operation of an optoisolator and how it could be used to have a computer light a fifteen watt light bulb to half brightness, he gave the reader almost no idea of how to determine if another type of device could be interfaced. How many houses are lit by fifteen watt light bulbs? He pointed out that only half of the ac signal got through when the optoisolator was conducting. Surely there are ways to get the whole waveform by using some sort of dual circuit. I am sure that it would have been worthwhile to include information on more useful things like 100 watt light bulbs, radios and burglar alarms.

Another unanswered question is how he came up with the value for his resistor and why it was connected two different ways in two different circuits. Another thing not mentioned was that the input diode could be connected with poles reversed to allow the circuit to be turned on with a positive voltage level. His elegant circuit to select one of ten lines and announce which was selected was nice, but I think that a latching interface to an I/O port permitting several lines to be enabled simultaneously would have been more useful.

In short, he gave enough technical information to allow copying of his designs with some understanding of why it works, but no idea of how to brew up custom interfaces. The things mentioned here were probably obvious to him. Otherwise he would have included them in the text. Unfortunately, the rest of us were left in the dark. His article was perfect discussion of optioisolators with an example of their use, but it falls far short of the title's promise, "Computer Control of The World".

Denis Bourdeau glossed over the possible problems with the SWTPC PR-40 printer in a way that left me more against the purchase than for it! With the types of applications I see for a printer, I became scared when he began to talk about not running the printer for more than one minute at a time. Although I have had no experience designing matrix print-heads, I do know that they are not designed to overheat after more than one minute's use. With the central component at stake, there should have been better assembly guidelines than "err on the light side of print intensity". I am happy for Mr. Bourdeau that all of *his* SWTPC kits worked the first time he applied power, but there are many who do not share his confidence.

Since September of 1975 I have been debugging a TVT II to be interfaced to a PDP-8. With a track record like that, I do not think myself ready to risk over two hundred dollars on something that threatens to burn itself up if I run too long of a listing. Perhaps I would have felt safer if he had given a more reliable method of approximating the ideal 400 μs pulse time. To me, it would be worth the extra money to buy a Digital Equipment DECWRITER which is fully assembled, tractor fed for up to a 132 column width, and will nearly last forever. How many overlong listings would you go through before having to buy a new print-head? In making many timing mistakes you will have practically paid for a DECWRITER. With printer technology constantly improving, I see no reason to risk buying the PR-40 unless I get more definitive resolutions for the problems associated with it!

Another, less important, example is found in Robert Grater's article on an interface between a TVT and a TTY line. I would have liked to have seen a table of other TVTs available that would work with his board. There are many TVTs available that are totally incompatible. There are those that act like a block of memory that just happens to print out on a TV screen. No mention is made of the fact that most TVTs available will *not* work with his unit. I would have liked a little information on his Bay Area TVT other than where I could buy one.

As a computer user, it is hard for me to give articles depth because it is necessary that I assume that the reader has a certain amount of knowledge of the topic and I do not always guess right. Most often, only one part of the problem is dealt with: the lexicon. It is easy to define words with words to make them look understandable, yet be meaningless. The definition must have sufficient depth so that the reader can see the theory, see an example of how the theory is used, then be able to relate it to what he thinks might be able to do with it.

William Cattey
Wallingford CT

REPLY FROM CHRIS BOWICK

Mr. Cattey, first let me say thanks for your letter. I really appreciate your comments as you have some very valid points. I will try to answer all of your questions, but not necessarily in the order that they appear in your letter:

1). The article does fall far short of the title's promise but, it is not the title I submitted for publication. The original title "An Inexpensive T^{2}L to Line Voltage Interface: The H74C1 Optoisolator" was apparently too dry for publication and was changed by the editors.

2). The intent of the article *as originally written* was *only* to *introduce* the H74C1 as a possible interface between TTL and line voltage operated devices. It was nothing more. Therefore, a "latching interface to an I/O port permitting several lines to be enabled simultaneously" would *not* have been appropriate and would have been far outside the scope of the article.

3). Remember, the article was intended (don't forget the original title) as a *very elementary introduction* to optoisolators. I was hoping it would stimulate interest and experimentation! That is the reason I included only very simple circuits. "100 watt light bulbs, radios, and burglar alarms," should be the subjects of a later article.

4). I have no formula for determining the value of R in the circuits. This is just the "typical" value (56k) which GE uses in its literature on the H74C1.

5). Thank you very much for pointing out the two different ways the 56k resistor was connected in the two different circuits (Fig. 4 and 5). This is indeed an error and is the one thing I had *not* intended to do. The 56k resistor in Fig. 4 should be connected as it is in Fig. 5. As simple as those circuits are, you would think I would have noticed before now. I guess it's just another case of not seeing the forest through the trees. I am truly sorry for the inconvenience.

6). If the input diode's connections on the H74C1 were reversed, nothing would happen! I'm sorry about that but it's true. Connecting 5 volts to pin 2 and the logic input to pin 1 would only cause the diode to remain reverse biased no matter what logic level was applied to pin 1. In order to allow the circuit to be turned on with a positive voltage, pin 2 would need to be grounded and the logic input applied to pin 1.

In closing, I again thank you for writing and I apologize that the title of the article did not indicate its contents. I will not let it happen again.

Chris Bowick
Atlanta GA

REPLY FROM DENIS BOURDEAU

My apologies for not going into greater detail concerning adjustment of the PR-40 electronics. It must be mentioned that the caution about print-head overheating only applies when driving the printer with the diagnostic software that SWTPC suggests. Once these adjustments are made, and when the printer is operated normally, little further concern about overheating is justified. Having used this printer quite heavily during the past half-year (having printed an estimated 50 thousand lines so far), one would think that low reliability is not a problem. Quite often, my listings will be 5 to 20 feet in length (that's 6 lines per inch) and even when making 20 foot memory dumps and software traces, the print-head gets no warmer than my coffee.

Remember that this device is available as a kit and SWTPC allows the experienced kit-builder to pocket the money that Heathkit puts into their "don't even have to know how to solder" documentation. When building digital electronic kits, it's recommended that one have an oscilloscope available; however, my scope never had to be turned on during this project. One might call it luck, but I call it mostly hard, meticulous work!

I've been asked about the noise generated during printing operations. I would say that the noise generated by the PR-40 is much more preferable than the "clackity-clack-clack-clacking!" produced by software-driven Selectrics I've heard. Remember that the PR-40 is perfectly quiet when idling – totally unlike most printing devices.

Two very minor hassles have surfaced with respect to PR-40 operation. (1) Its dependence on 5/16" ribbon motivated me to convert it to use 1/2" ribbon available at many grocery stores. (2) The paper tends to wrap back around the platten near the end of a roll (due to the tight curvature near the end). Problem 2 is solved by the trivial addition of a case or flat plate that the paper would be guided by. (With all my other projects competing for my time, a piece of cardboard taped into position satisfies me for now.)

While this printer possesses many limitations when compared to a DECWRITER, for example, the function it performs is much more important to me than *how* it performs its function. These limitations are particularly serious if commercial applications (i.e. business report generation) are intended. In this case, however, we're not talking about the practical hobbyist's concerns.

There is an important idea that one must always remind oneself about. This idea has most relevance to the hobbyists and other people involved with practical computer applications. The idea was best stated by a friend, who more wisely than eloquently, said, "It's not what ya got; it's how ya use it!"

Denis Bourdeau
Littleton CO

REPLY FROM BOB GRATER

In Reference to Mr. Cattey's letter: The SAB-1 was meant to be strictly a Serial to Parallel converter, and possibly I could have stated such more explicitly – and as such it will interface to parallel ports on TVTs, line printers, converted Selectrics, etc. – any device that requires parallel ASCII input. The Bay Area TVT was an example that happened to be handy. Other interfacing that I personally know of are into a SWTP CT-1024, Central Data TVT, and Beehive terminal. I think that Mr. Cattey would be more correct in stating that 'most *video boards* available will not work with his unit', there is a big difference between a stand-alone TVT and a video board for computer plug-in.

The listing as to where the Bay Area TVT is available is shown at the end of the article and attached is a flyer that Byte Shop #2 puts out giving information on the TVT. I felt that, to do any justice to the Bay Area TVT would require a separate article and anyone requiring the information could write to them for it.

It's a common failing of most technical writers (especially me) to assume too high a theory baseline when writing an article – not because we are any more knowledgable, but because we are intimately familiar with the subject we are writing about. So, Mr. Cattey has a good point, and a reminder now and then sure helps keep my feet on the ground!

The Bay Area TV Typewriter is starting to become known nationwide, and Byte Inc. has now improved the documentation and packaging (of course raising the price). The Bay Area TV is a 32 character by 16 line, 2 page, parallel input board, with full cursor control, roll, and scrolling. It requires either a video monitor or rf modulator for display, and ASCII keyboard or parallel computer input. Power requirements are 5V @ 1.5A and -12V @ 50mA. This unit, when used with the SAB-1 board and an ASCII keyboard, gives you a full terminal to access the TTY port on KIM. Some of the lesser known features of the TVT are Direct Memory Access and a timing output that will give you 110 baud thru a divide by 9 counter for driving the SAB-1 or a modem. The PC board alone sells for $35.00, and the complete kit with documentation runs $139.00 which also includes a copy of Don Lancaster's "TV Typewriter Cookbook". Please add $2.00 for shipping, and the documentation alone costs $2.50.

Bob Grater
Santa Clara CA

LACK OF CONFIDENCE

I would like to comment on the article "Chasing Those Naughty Bits" by John Molnar. This program is similar to the ROBIT memory diagnostic I received with my Southwest Technical Products M6800 computer. Rather than providing the user with the confidence in the memory system that Mr. Molnar claims, this test will not reveal a significant failure mode in memory boards. A failure of the memory ICs or address decoding circuitry can cause two or more addresses to respond to the same address. Thus when the test routine stores a pattern in location X the same pattern or part of it may store elsewhere. This type error is revealed by diagnostics that store different test patterns in ALL memory locations and then test to see if the memory has retained the pattern. This is the type of program used in the second diagnostic supplies with the SWTP6800. Both tests would have to run for me to have confidence in a memory system.

An additional comment should be made about "Branch to self" command given in the text of the article. 20FF will not branch to itself. In fact it will store the index register in some erroneous location and continue on. A correct branch to self is 20FE which I have used often.

Leo Taylor
West Haven CT

As far as the first point about the test program, you're correct, Leo. However, the test was not designed to check multiple addressing (hardware) and was indented as a "simple" test, as the article implied. Memory tests can go on and on, and can get to a point where it is not practical to key in the test by hand (as mine were). So, you have a point, but not an error.

The second point is MY fault – didn't proof the author's proofs well enough. Indeed, the 20FF should be 20FE. What can I say? – John Molnar.

(As a point of interest, and since he is one of our best writers, John Molnar recently joined the staff of 73 as Executive Editor. Why can't I get a title like that?! Hopefully, we'll still be seeing some of John's material in the future in spite of the fact he's probably working harder than he ever has. – John.)

Feb. 5, 1977

1821 Overglen Dr.
Plano, Texas
75074
JC

Kilobaud

Dear Mr. Green,

My name is Michelle Parks, I am 8 years old.

Uncle Don gave me a subscription to your magazine for Christmas; but, I have not received any copies yet.

Uncle Don says I will grow up to be a tax collector with out Kilobaud.

Please check on my subscription.

Thankyou

Michelle Parks

Plano TX 75074

We checked . . . we straightened it out . . . and, we saved you from a horrible fate. – John.

FOR A QUICK STOP

This is just a note to tell you how much I enjoy *Kilobaud Magazine*. You are really packing the information into the pages!

One article that caught my eye in the latest issue was the one about the paper tape reader, by Dr. Doug Hogg. He uses some very sound design practices in the reader and, if you twisted my arm enough, I could believe that he has copied the basic concepts of the Chalco Model 5000 series of readers.

The Chalco readers are high-quality, high-speed (625 CPS) readers which use a capstan/pinch wheel combination to pull the tape through the read head. They use two magnetic brakes to halt the tape and are capable of *stopping the tape on a character at a speed of 625 CPS!* This stopping capability brings me to the point of this letter.

As Dr. Hogg mentioned in the article, the tape must stop within 1 ms at a speed of 500 CPS. The following specifications were taken from the Chalco Model 5000 manual: Stop time: less than 200 microseconds; Brake coil current: 2 Amperes; Brake coil resistance: 0.38 Ohms; Brake coil inductance: 3.6

millihenries.

As you can see, this brake coil is no ordinary magnet! In addition to the electrical specifications, the flapper plate is extremely close to the magnet to minimize the travel time.

Perhaps this information will be of value to anyone desiring to build a tape reader as Dr. Hogg described. By the way, the Chalco uses two brakes, one on each side of the read head. This reader is a real gem and anyone finding one on the market would be well advised to invest the hundred or two dollars to buy it.

Michael M. Dodd
Fairfax VA

FRANCHISE

Let's see an article on microcomputer store franchises. You can do a better job than the Wall Street Journal on the subject. As a buyer of services, I would like to know what they have to offer. I'd also like to know what the costs are to get a franchise.

Paul Sternberger
East Brunswick NJ

Okay, I think a lot of people would be interested in this. I've written to Imsai and MITS asking for information on their franchise plans and I'll try to pass it along as soon as I get answers from them. – Wayne.

1 Print "Publisher's Remarks" 10 No End Run

from page 2

Atlanta, June 18-19. Computerfest. Contact Atlanta Computerfest, 53 Old Stone Mill Road, Marietta GA 30067.

Seattle, July 29-31. Computerfest/Hamfest. Contact Harry Lewis, 10352 Sandpoint Way, Seattle WA 98125. (206) 523-9117.

Albuquerque, May 4-7. Computerfest (MITS WACC). Contact Charles Olsen at MITS.

HARDWARE & SOFTWARE REVIEWS

from page 5

run on a Z-80 without program modification.

Another minor difference between the Z-80 and the 8080 is the fact that the Z-80 does not place status and control information (in the form of certain signals) on the data bus at the beginning of each machine cycle (SYNC time). The final item (with regard to differences) lies in the fact the Z-80 will not stop at M1 cycle time consistently when used with hardware front panel systems. To overcome the possibility of having it stop on a random cycle a simple cure is to hit reset after stopping.

The Advantages of the Z-80 (and the DZ80-80)

There have been several articles describing the Block Move, Block Search, Bit Manipulation and other instructions which make the Z-80 more desirable than the 8080. There have also been descriptions of the dynamic refresh register built into the Z-80 and how it makes use of dynamic memory so practical. Therefore, since my objective in writing this review was not to point out the advantages of the Z-80, but instead the advantages of Dutronics' DZ80-80, I won't be getting into the finer features of the chip itself.

The software package provided by Dutronics with their DZ80-80 is impressive! They include a complete monitor system on paper tape along with the board (at no additional cost). Table 1 lists the thirteen commands available with this package. The software has two I/O structures: *The Command I/O* specifies and handles the device from which system instructions are received and the *File I/O* handles serial data to and from such devices as cassette recorders, TTYs, paper tape readers and punches. The Command and File I/O port assignments can be changed after loading the monitor into your system.

My first attempts at loading the monitor met with failure so I put in a call to Dutronics and the president of the company, Dave Dutra, helped me track down the problem. It turned out to be a hardware failure in my computer rather than his software (which made him feel good, and made me feel bad). The monitor program has since been loaded into a 2708 EPROM and is essentially being used as a front panel replacement for my Imsai.

Conclusions

All I can say at this point is, "Do it the easy way." You can convert your 8080 CPU board to a fully operational Z-80 system for $159.95 using the Dutronics DZ80-80. The price is right, the board does the job and the company is a pleasure to do business with.

Dave Winthrop
Santa Maria CA

Around the Industry

from page 16

delay distortion is relatively unimportant. In the first waveform of Fig. 1, the logic Zero state is defined by a half cycle of the low frequency (1200 Hz) signal. Because of the phase shift and group delay distortion existent in the portable audio cassette recorders, this half cycle component causes substantial "bit shifting" jitter when the waveform is played back. In the PerCom Cassette Interfaces at 1200 Baud, a full cycle of 1200 Hz defines a logic Zero and the "bit shifting" is substantially reduced. This shifting is easily observed by studying the "eye" pattern on an oscilloscope. As long as we wish to use readily available audio cassette recorders to store our programs, it would be a mistake to make PE 2400 a standard. Obviously this conclusion does not apply to cassette drives designed for this waveform.

Second: Mr. Walker advocates one stop bit. While it has very little effect on data recovery reliability, a single stop bit provides no protection against "overspeed." Suppose a cassette is played back on a player running 4% faster than the recorder on which it was recorded. Further suppose that while loading the recorded program into the computer we wish to display it on a data terminal. The reliability of the data going into the computer will not be affected since the system is self-clocking. However, the 4% cassette overspeed means we will try to send 26 character to the terminal in the time interval when it can only accept 25 character. This means the UART interface driving the terminal will not be ready to accept one of every 26 characters. If the data had been recorded with 2 stop bits, the terminal read back would be able to tolerate a 10% overspeed. Again, this does not affect the reliability of the data going to the computer, but it does make visible replay possible if the play back cassette is running fast.

Final Point: Concerning Mr. Walker's opinion of the parity bit; not all recording is ANSCII characters. If the data is recorded 8-bit Binary there is no room in the 8-bit format for a parity bit. Furthermore, since the kind of error encountered on cassette playback are usually "Burst errors" in which multiple bits are destroyed, a simple parity check has only a 60-80% probability of recognizing an error. Consequently I feel "in-character" parity should be left to the specific application and a longitudinal checksum is to be preferred as a matter of practice, but should not be imposed as a uniform standard requirement. Most of the commonly used Hex and Binary loader formats include a checksum although the summing algorithm varies.

From Clint O'Connor, a reader . . . I was somewhat confused by Hal Walker's discussion on checksums in his "Kill a Byte" article of February, 1977. He mentions that each company has its own checksum procedure and that you must know the sum. I am not familiar with checksum procedures as applied to magnetic tape but they are quite simple insofar as paper tapes are concerned, and there is no need to know the checksum number. One simple way is to add every byte punched on a paper tape to a 16 bit accumulator, ignoring overflow conditions. At the end of the punched tape, you simply punch two more bytes representing the contents of this 16 bit accumulator. When reading the tape back, whether it is yours or someone else's, simply sum up all the bytes read off the tape as you store the data into memory. At the end, retrieve the last two bytes stored and subtract this 16-bit checksum twice (you've accumulated all the data bytes *plus* the punched checksum). If the result is non-zero, an error has occurred. Parity bits are not desirable on paper tape, since using a parity bit leaves seven other bits on 8-channel paper tape. Representing a full byte with 7 bits is awkward, to say the least. However, parity can easily be applied to magnetic tape storage. My point, though, is that checksum procedures should be relatively easy to apply to magnetic tape storage. It is quicker to add a byte to an accumulator during a read operation than it is to check the parity. This does not apply to hardware parity checkers, however. I believe the best standard is the simplest one, and checksums offer little software and lower cost (zero) compared to fast hardware parity checkers.

20 Print "Editor's Remarks"
30 End
Run

from page 3

after seeing documentation available for building some of the kits which are presently available I can appreciate this attitude. Anyone who has ever built a Heathkit has a pretty good idea of what they will be offering. I just recently finished construction of a Heathkit Telephone Amplifier which I'm very happy with. And, true to form, every time I tried to anticipate what was coming during the construction steps, and deviated, I found myself in trouble and having to go back and correct what I had done. They really do have their act together.

Looking Back – May 1975

As evidenced by a 72-page issue of the *Micro-8 Newsletter* which came out in May '75, there were hundreds of people doing a lot of interesting things. Unfortunately, there weren't any really significant events taking place. But, we were certainly leading up to it . . . the following month (June '75) two of the largest computer clubs in the country were formed.

Hal Chamberlin presented the second part in a series on cassette standards in the May issue of *The Computer Hobbyist.* Oh yes, and I found an interesting tidbit in that issue of TCH. In an advertisement Bill Godbout mentioned that he was the first to offer the 8008 to hobbyists "over 16 months ago" and he was conducting a contest to see who could guess "the Secret Microcomputer Company." The ad ran ". . . now we're setting the pace again with a powerful new 16 bit microcomputer IC in a 40 pin DIP . . ." and readers were invited to guess the manufacturer and write in 25 words or less why they should have one of these chips. (There were three winners to that contest. Who were they, and what are they doing with the 16 bit microprocessor chips they won? Come forth and be identified!) I never realized the blatent hint Bill included in that contest . . . "setting the pace again. . ."

In May of 1975 there were five publications serving the computer hobbyist community. These were *The Micro-8 Newsletter, The Computer Hobbyist Newsletter, The People's Computer Company Newsletter, The Homebrew Club Newsletter* (Issue #3 – they were up to 80 members) and a newsletter put out by the now-defunct MP Publishing Company in Bedford, Massachusetts (I don't know what happened there . . . I guess he just couldn't keep the thing going). Information was scarce in those days, and with only four significant publications to gather data from, most hobbyists were like sponges when they got their hands on one of them.

Speaking of scarce items . . . it seems there was somewhat of a shortage of I/O boards to go with the Altair 8800s which were finally being shipped then. This didn't bother Steve Dompier up in San Francisco! He sat down and wrote an unusual program for generating music with his Altair. The unique part is that the music was played through a regular AM radio placed next to the computer. (He wrote an article on the program and technique for the *People's Computer Company Newsletter* back then and he'll be doing an update on it for an upcoming issue of *Kilobaud.*)

Miscellaneous

Several years ago I had a book in my library which stood out among the rest as far as being a fantastic reference. Somebody else with good taste in reference books came along one day and decided they needed it more than I did. I've been trying like crazy to find another copy of it ever since then. The other day it occurred to me that I could simply ask 35,000 other people for some help in locating another copy. (Ahh, the benefits of the job!)

The book was a paperback (about 10" by 7") version of a study guide for the Certificate in Data Processing Examination. I think it was dark green. Any help would be appreciated . . . thanks.

NEWS OF THE INDUSTRY

from page 8

new drive and the unique *uniboard* method of construction make it possible to produce a data cartridge recorder for about one fourth the normal price of such recorders. Operates in the self-clocking Phase Encoded mode at 9600 baud or less. Motor and record/play functions are under keyboard or software control. Overriding pushbuttons make manual operation possible. Readily interfaces with commonly used microprocessors and minis with serial I/O ports.

Available also in export version. Price includes read/write electronics Phase Encoder board (16 X or 1 X clock) and motor control electronics. Delivery 4 weeks ARO. Price $220.

For further information contact National Multiplex Corporation, 3474 Rand Avenue, Box 288, South Plainfield NJ 07080.

TRENTON COMPUTER FESTIVAL

The second annual Trenton Computer Festival (April 31–May 1) will be bigger and better than the original. It is being expanded to two full days, with new and larger facilities to house up to 90 exhibitors.

Computer conference sections and forums are planned on the following topics: Microcomputers for home, radio amateurs, education and medicine; consumer applications of microprocessors; computer music; robots; graphics; speech synthesis; establishing amateur computer standards; computer club convention.

For further information call Jaci DiPaolo (609) 771-2487.

BOOKS

from page 13

logic family available to experimenters is CMOS, short for Complementary-Metal-Oxide-Silicon. This family provides several important advantages when compared to TTL. Power consumption is very low, eliminating the familiar bulky TTL power supplies. CMOS is also immune to much of the noise that plagues TTL circuits, and power supply voltage is not critical. In short, CMOS is enjoying great acceptance by the hobbyist market, due in part to its wide availability, low cost, and versatility. And, as expected, Don Lancaster has again responded to the experimenter's needs with his latest offering, the *CMOS Cookbook*.

The *CMOS Cookbook* is patterned after its TTL-oriented predecessor. An introductory section provides insight into the design and operation of CMOS itself, and the features and disadvantages of the logic family are described. Power supply and mechanical considerations are discussed in a manner easily understood by the casual experimenter. The most popular feature of all the *Cookbooks* has been retained – the chapter on pinouts and characteristics of the available chips. This section in my *Cookbook* has already become ragged through excessive use.

A description of general logic conventions is provided (a review for earlier *Cookbook* enthusiasts!), as well as a discussion of basic gates and circuits. Clocked logic, multivibrators, and counters are treated in following chapters. Lancaster's approach should appeal to the novice as well as the seasoned experimenter. Each chapter introduces a new technique, and the practical design considerations relating to the new subject are discussed. Actual circuits, complete with component values, are then presented. A multitude of practical circuits, such as music generators and voltage controlled oscillators, are described for the serious experimenter.

The *CMOS Cookbook* is a must for the digital hobbyist. This reference contains virtually everything required to "get into" CMOS design. And, it is the only reference I have encountered that is able to be read from cover to cover without losing its appeal. It requires a logical, well-organized reference to make this possible. The home computer enthusiast should not be without a copy, as most of *today's* microprocessor systems are based on MOS logic.

John Molnar
Executive Editor
73 Magazine

The BASIC Forum

from page 17

think the ON INTERRUPT statement is an idea whose time has come. Anyone have ideas for implementation?

A local friend, Bob Strader, 2347 Alta Mira, Tyler, Texas 75701, has a few ideas for BASIC statements that we thought might be useful. Bob is particularly interested in game programs – not ones in which you play the computer, but rather games in which you match wits with another of your own species. A statement he has suggested has the following form:

```
ECHO ON

or

ECHO OFF
```

The ECHO statement could be used to turn the output device (TTY or CRT) on and off during INPUTing. In the ECHO OFF mode, it would permit certain data entry by game players to be privileged – at least to the extent that the player could conceal what he is doing at the keyboard. With the ECHO ON command, other entered data could be available to all onlookers. Bob has provided several suggested games that would use this capability to advantage. Others interested in human versus human games for computers can write Bob at his address above.

A final suggestion for improvements to BASIC comes from another high school student, David Piper, 212 Tamiami Trail, West Lafayette, Indiana 47906. David wants to see some BASIC statements for handling page mode CRTs. David has done quite extensive work in the area of animated CRT games, but finds it difficult to exchange programs because of the peculiar aspects of his page mode terminal. David's dilemma may not have a simple solution. He has suggested one new BASIC statement that would help.

```
HOME

or

HOME(ERASE)
```

The HOME command would move the CRT cursor to the upper left of the screen without modifying the contents of the display. The ERASE option would clear the screen in addition to homing up. Often, repositioning of the cursor to specified points on the screen is required. Another new BASIC statement might be created for this purpose. It could have the following appearance:

```
CURSOR(L,P)
```

When executed, the cursor would move to line L and position P.

In this month's BASIC FORUM, we have just scratched the surface for ideas concerning new BASIC statements. Perhaps you have some too. Send them with any other comments you may have to

THE BASIC FORUM
c/o Dick Whipple
305 Clemson Drive
Tyler TX 75703.

Commands	
DATA	ON-GOTO
DEF	ON-GOSUB
DIM	OPTION
END	PRINT
FOR-NEXT-STEP	RANDOMIZE
GOSUB	READ
GOTO	REM
IF-THEN	RESTORE
INPUT	RETURN
LET	STOP
Functions	
ABS	LOG
ATN	RND
COS	SGN
EXP	SIN
FN-User defined	SQR
INT	TAB
	TAN

Fig. 1. List of commands and functions contained in the ANSI Minimal BASIC Standard.

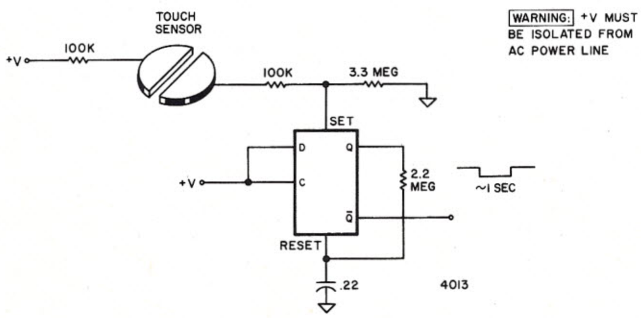
```
 10 REM:PROGRAM START
...
100 PRINT "AGAIN";
110 INPUT A$
120 IF LEFT$(A$,1)="Y" THEN 10
130 IF LEFT$(A$,1)="N" THEN 160
140 PRINT "PLEASE ANSWER 'Y' or 'N'."
150 GOTO 100
160 END
```

```
 10 REM:PROGRAM START
...
100 PRINT "AGAIN";
110 ANSWER 10,120
120 END
```

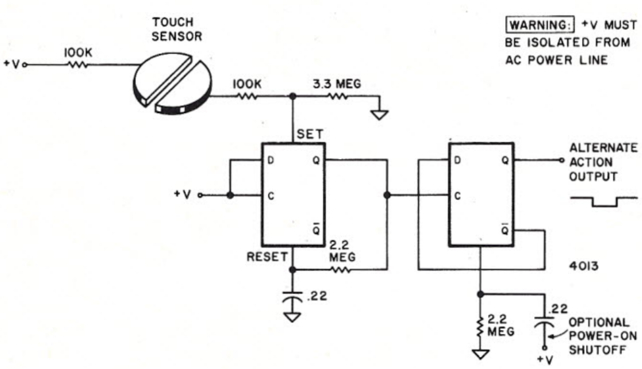
Fig. 2. Two sample programs by David Price showing a typical application of his proposed ANSWER command and the advantages it offers.

Don Lancaster
SYNERGETICS

Clocked Logic



(a) Pulse output.



(b) Alternate "off-on" action.

Fig. 15. Touch switches based on conductivity.

In this third and final article in his series on flip-flops, Don Lancaster presents some of the most interesting and unusual applications we've seen in some time. Most of the circuits deal with some form of A-to-D and D-to-A conversion and are made possible by the electrical characteristics of the Complementary Metal Oxide Semiconductors (CMOS) circuits used throughout. As a matter of fact, the material has been extracted from his upcoming "cookbook" to be published by Howard W. Sams . . . CMOS Cookbook. – *John.*

The nearly infinite input impedance of CMOS makes it ideal for use in *touch* or *proximity* circuits. Usually a touch sensitive circuit needs physical contact, while a proximity circuit only needs a nearby presence.

There are several characteristics of the human body that may be used for touch sensors. Human skin resistance is usually several hundred thousand Ohms, but varies with the individual, the contact spacing, and physiological factors. The capacitance of the human body is usually around 300 picofarads referenced to ground. Touching a point in a circuit usually has the effect of adding 300 picofarads of loading to that point. Just coming near a sensor provides a capacitive divider set by the 300 pF and the capacitance between sensor and person. Usually this is only a very few picofarads and drops dramatically as spacing is increased.

Finally, the human body acts as an antenna, picking up the 60 Hertz power line near field in virtually all indoor and most outdoor situations.

There are several important things to watch for if you want a reliable touch sensor. The sensing circuit works best if it is solidly grounded. Portable or ungrounded circuits may not work at all unless they have a reference against which to work.

Hot chassis techniques of

. . . Part 3:

Data Converters and Special Functions

any kind where the sensor returns to one side of the power line should, of course, be avoided entirely. Even if ultra high series resistors are used, what ends up as an unnoticeable leakage current to one person can end up as a mild shock or at least a "liveness" or fuzziness sensation to others.

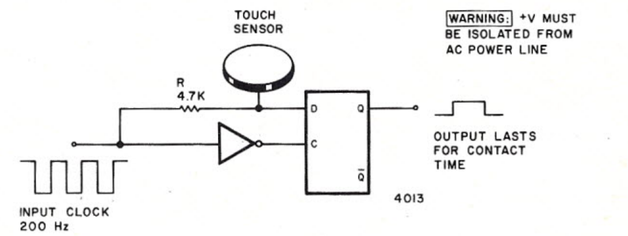
Most touch sensors should be debounced with a long time constant, preferably a second or so. This prevents any nervousness or hesitancy from being entered as multiple hits. Always use only the absolute minimum sensitivity you need for any touch sensor and provide as much protection as possible to the sensor. This minimizes both static damage potential and possible false alarms from power line transients, AM radio signals, and so on.

Let's look at some examples. Fig. 15 shows us two touch switches based on conductivity. Fig. 15(a) illustrates the triggering of a monostable via the Set input when the touch sensor is bridged with a resistance that is small compared to 3 megohms. A conditioned output pulse one second long results.

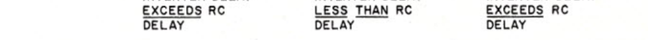
We can convert this to an alternate action off-on circuit (Fig. 15(b)) by adding a binary divider to the output, using the other half of the 4013 with its $\overline{Q}$ output cross coupled to D. The final resistor and capacitor are an optional external reset that makes sure the sensor comes up in the off state.

A capacitance operated touch system for electronic music keyboards is shown in Fig. 16. An input square wave is used as a clocking signal. Normally, the inverter delays the clocking till *after* the D input has accepted a new value, so an output low results. Touching the D input adds 300 picofarads or so of capacitance to ground. This gives us a resistor-capacitor delay network that slows down the waveform reaching the D input. In fact, it slows things down so much that the clock gets there first, giving us an output high condition. The choice of clock frequency sets the debouncing you'll get. Always use the lowest possible frequency for any particular use. For many electronic music uses, 500 Hertz or higher is a good choice.

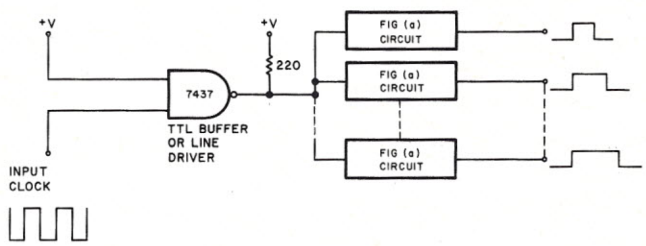
Note that our series



(a) Basic circuit — input clock frequency sets debounce time. R sets sensitivity.



(b) Waveforms.



(c) Input buffer minimizes interaction between multiple contacts.

Fig. 16. Touch switches based on capacitance have many electronic music uses.

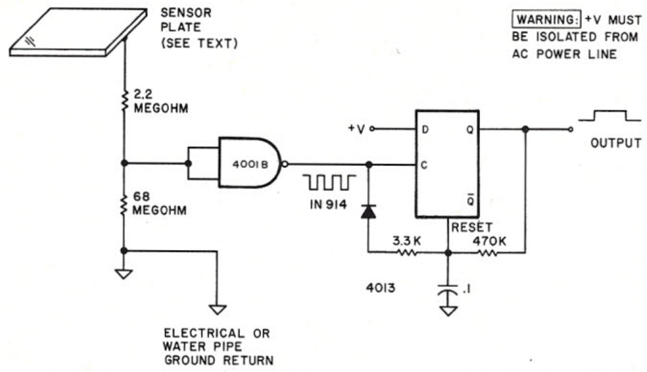
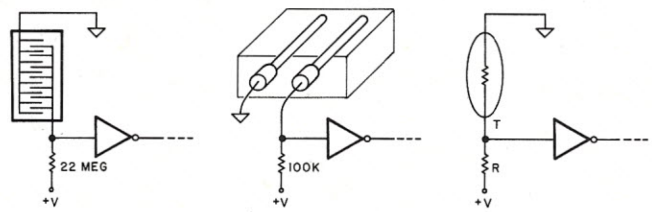
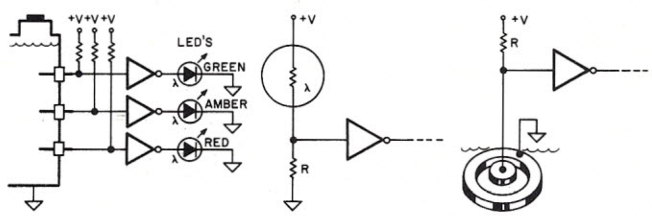


Fig. 17. Proximity switch based on power line "hum" coupling.

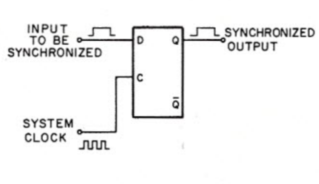


(a) Rain sensor. *(b) Nutrient saturation (hydrophonics).* *(c) Thermistor or thermostat.*

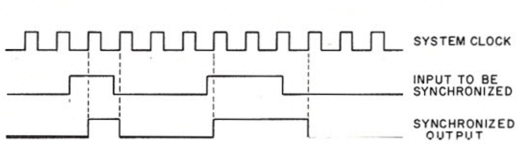


(d) Booster tank level (firefighting). *(e) Photoconductive cell.* *(f) Splash alarm (swimming pool safety).*

Fig. 18. Other high impedance conductivity sensors such as these may *be directly CMOS compatible. Unless conductivity change is constant, sudden, and dramatic, an input CMOS operational amplifier is recommended for conditioning and adjustment.*



(a) Circuit.



(b) Waveforms.

Fig. 19. The synchronizer circuit is used to lock erratic, delayed, or outside-world signals to system timing.

resistor is somewhat low in value. This is used to set the sensitivity. Since this is a fairly low impedance point, hum and interference pickup is unlikely as is finger-to-finger conductivity when multiple keys are hit.

In multiple key systems, such as Fig. 16(b), some means of keeping the fingers from loading the clock source rise and fall time must be provided. You can do this with an extra inverter or else you can provide an exceptionally stiff (low impedance) driver for the entire circuit. A TTL line buffer is a good choice.

Fig. 17 shows us a proximity switch based on human coupling of the 60 Hertz power line. A hand very near the plate induces hum into the first gate. This is squared up and used to trip the retriggerable monostable as shown. A clean output results from the instant of first proximity till a few milliseconds after release. The sensitivity depends on the size of the plate and the amount of allowable false alarms from induced noise sources.

Some Related Sensors

The thing activating the resistance style sensor of Fig. 15 does not have to be a people; any high impedance that can't be heavily loaded will work as well. Fig. 18 gives several examples. The rain sensor of 18(a) uses the conductivity of rainwater to conduct across a sensor grid. Two probes in the bottom of a hydrophonic greenhouse bed in 18(b) will conduct when nutrient solution reaches them. A thermistor or thermostat is shown in 18(c). As the resistance drops, the output of the inverter will go high. A liquid level system suitable for the booster tank on a fire engine is shown in 18(d). A photocell system is shown in 18(c), followed by a splash alarm for swimming pool safety in 18(f).

One thing is essential for all these applications to work. The difference between the OFF and ON resistances has to be very large and very stable. If the resistance drifts with time or a sharply defined threshold is needed, you'll have to add an input conditioning circuit, perhaps using a 3130 op-amp.

Synchronizers

Synchronizers are a very important class of circuits. They are used to lock outside world signals (or other signals that are random or somehow out of step) to system timing.

Our basic synchronizer connection for the 4013 appears in Fig. 19. The signal to be synchronized is applied to the D input. A system clocking signal goes to the clock. The locked signal output is taken from Q. As the waveforms in Fig. 19(b) show us, both the leading and trailing edges of the input are delayed until the next positive clock edge. The output locked signal will be a delayed replica of different width than the input, but the output will always be locked to the clock timing. The output will also always be an exact number of clock intervals in width.

For some synchronizer uses, we can refer to Fig. 20. In 20(a), we use a synchronizer following a memory or character generator. The synchronizer samples and holds the data only when it is valid, relocking the outputs to system timing. This eliminates possible times when the memory is accessing, settling, or otherwise putting out garbage. What we have done is convert a short mixture of good and bad data for an always-accurate answer delayed one clock pulse interval in time. In 20(b), we make a ripple counter look like a synchronous one by decoding a state *one count early* and relocking it to the input clock. This eliminates any ripple delay and propagation times. In 20(c), we've used a synchronizer to eliminate the "inherent" one count bobble or ambiguity of a frequency counter. We do this by delaying the counter's time reference gate until a pulse to be counted gets there. A whole number of input pulses are always counted this way. This eliminates the random starting and stopping points of the time gate and the apparent one count bobble.

One and Only One

The one-and-only-one is a very important synchronizer circuit, detailed in Fig. 21. It will give you exactly one clock interval as an output in response to an outside world command. Our outside world command is shown as a positive edge. This sets the first flip-flop. The first flip-flop absorbs the time difference between the arrival of the outside world signal and the system clock edge. Our second flip-flop generates a one clock wide output pulse and resets the first stage. Every time you outside-world trigger the circuit, you get one and only one clock pulse interval as an output. Waveforms are shown in 21(b).

The one and only one may be used as a single frame update generator for a TV Typewriter per Fig. 21(c). The positive edge of an "enter" command from a keyboard is used to direct set the synchronizer. An output one vertical interval (16.7 milliseconds), and lasting one frame, is provided. This circuit also gives us a repeat option. If we make D high and put a low frequency (7.5 Hertz) *blinker clock* on the first stage clock input, we get repeated single frame outputs reoccurring at the blinker rate. For normal operation, hold D low; for repeat make D high.

You can convert a one and only one circuit into a N and only N by adding a counter that delays resetting the first stage until N counts have gone by. A gate may be added to make this circuit into a generator of N clock pulses.

Two Sequential Circuits

Two sequential circuits that use 4013s are shown in Fig. 22. The *bucket brigade* of Fig. 22(a) sequentially gives us self-decoded outputs as shown, first at A, then at B, then C, as shown. Only one start command must be provided per sequence if you are to avoid multiple outputs. You can close the circuit on itself for a continuous "electronic stepper" operation, but you'll have to add some method to make sure only one output is high at a time.

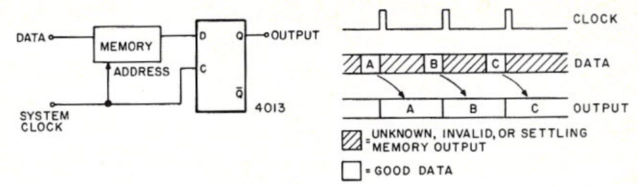
The *sequential pass-on* of Fig. 22(b) is handy for alarms, electronic locks, and other circuits where events must follow each other in a certain order. You get an output only if the leading edge of event A happens and is followed by event B and then event C. Events in the wrong order or not present prevent an output. Since the event inputs all go to clocks, they must, of course, be properly conditioned, bounceless, and noise-free.

A Programmable Divider

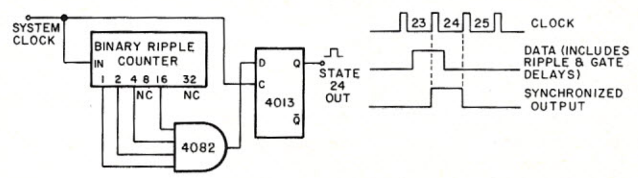
An interesting if somewhat oddball use of a 4013 is the programmable divider of Fig. 23. The circuit will divide a clock either by one or two. This is handy in digital data recording circuits where a string of clock cycles related in frequency by 2:1 may be needed. With the *Mode* input high, the 4001 NOR gate is disabled and the 4013 acts as a binary divider. We get the binary division through the $\overline{Q}$ to D external feedback connection. If you ground the Mode input, the leading edge of the clock sets the 4013 and the trailing edge resets it via the direct set input, all in one clock cycle. For proper operation, the RC network has to have a short time constant compared to twice the clock frequency. The clock input must have a symmetrical (50-50) duty cycle.

Digital-to-Analog Converters

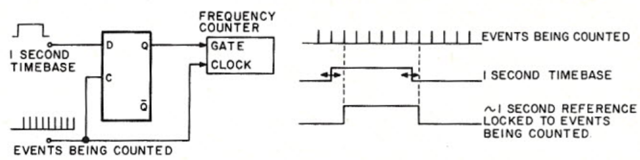
Most digital-to-analog con-



(a) A synchronizer following a memory samples the memories output only when data is valid and stable. It saves the good output for the entire next system clock cycle.

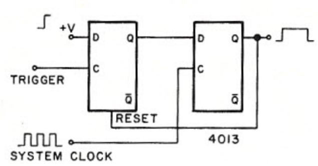


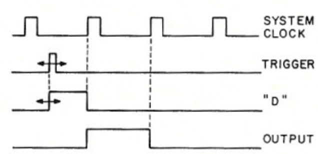
(b) State #24 of this counting chain is synchronously decoded and output by ripple decoding state #23 and resynchronizing. This eliminates ripple delay times.



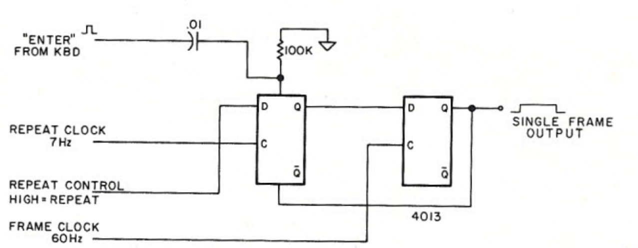
(c) The one count ambiguity or bobble of a frequency counter may be eliminated by delaying the time reference so it starts and stops on an event being counted.

Fig. 20. Using synchronizers.



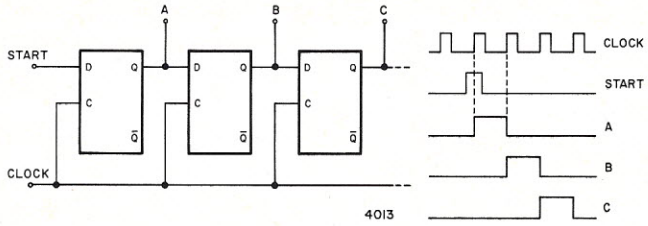


(a) Circuit. *(b) Waveforms.*

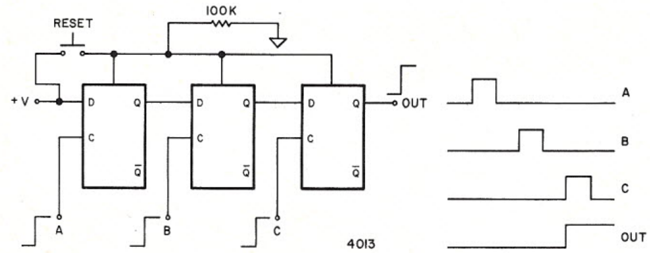


(c) Single frame update generator for a TV typewriter.

Fig. 21. The one-and-only-one.

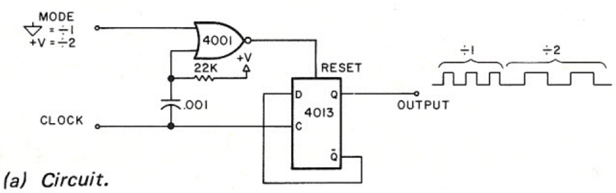


(a) The bucket brigade synchronously self decodes.

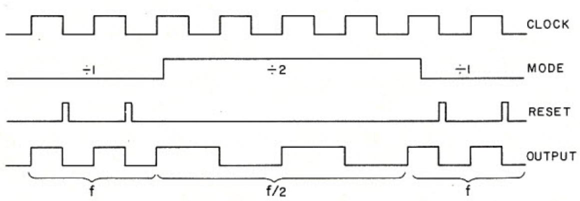


(b) Sequential pass-on provides an output only if C follows B follows A.

Fig. 22. Sequential circuits.

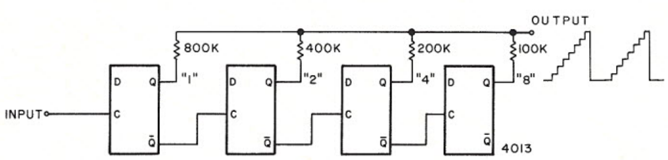


(a) Circuit.

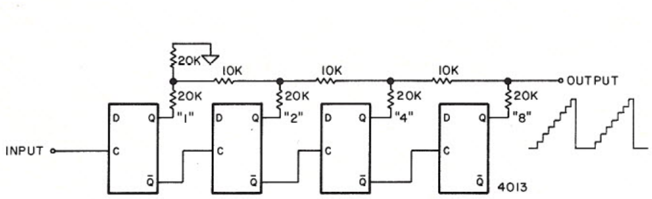


(b) Waveforms.

Fig. 23. Programmable divider divides by one or two.



(a) Current mode uses "1-2-4-8" network. Low output impedance and voltage;



(b) Voltage mode uses "R-2R" network. High output impedance and voltage.

Fig. 24. Digital to analog (D/A) converters.

version schemes are based on digitially switching current sources, voltages, or resistors and then summing the resulting current as an analog output. CMOS output stages are especially attractive for this because they swing the full supply range and have no offsets.

Fig. 24 gives us two different digital-to-analog conversion schemes. We've shown the digital portion as a binary divider using 4013s. The counter is advanced to the desired count and then output as an analog voltage. As an alternate, we could use our D-flops as latches and apply a parallel digital word. The word is updated every time we want a new analog conversion.

The two circuits differ in our choice of summing resistors. In Fig. 24(a) we use the classic "1-2-4-8" resistor weighting method. But note one important detail: The current through our resistors is inversely proportional to resistance, so the 800k resistor is weighted 1 and the 100k resistor is weighted 8.

Traditionally, this circuit summed into the virtual ground summing input of an operational amplifier. But thanks to CMOS being able to identically source or sink current to either supply rail, our output loading can be anything from an open to a short and can be returned to any voltage between the supply limits without affecting resistor values or linearity. (You can prove this hard-to-believe bit with Thevenins theorem; the "source impedance of our multiple supply is constant; only the equivalent "single source" voltage changes.)

For more resolution, the resistor values tend to get out of hand. We can combine the resistors with a current or voltagydivider circuit and eliminate this restriction. This is called the "R-2R" method and is shown in Fig. 24(b). Outputs of both circuits are the same, but the R-2R circuit easily expands to long word lengths. Another of its advantages is that the source and sink currents of each stage are identical, so any internal drops tend to compensate themselves rather than getting progressively worse.

You can make these circuits into analog-to-digital converters by insiding them out. The analog output goes to a comparator that starts or stops the clock pulses depending on an input analog signal. The digital output is then taken from the counter.

Phase Shifters

Digital *phase shifters* may be used to generate various delays in a signal. This is handy for color TV processing, audio filtering, radar signal correlation, and similar uses. The same idea is also used to generate multiphase ac power source signals and the multiple clocks needed for some microprocessors. Fig. 25 gives us some details.

The binary divider of 25(a) is the simplest example. It starts with a double frequency clock and gives us two clock phases that are spaced 180 degrees from each other. If some "daylight" is needed between clock phases, a narrow clock pulse can be ANDed with the outputs to provided a zero state between the phase outputs.

In Fig. 25(b), we generate two *quadrature*, or 90 degree phase shifted, digital signals. We start with a 4F clock and use an even length walking ring counter to generate the phases for us. Unlike audio networks, the 90 degree phase shift is independent of frequency and follows the input clock over any desired frequency range. By decoding both-outputs-high and both-outputs-low, we can generate the type of two-phase clock signal needed by a microprocessor where one clock goes high, followed by daylight, followed by the other clock high, followed by more daylight.

Incidentally, its extremely important in many electronic circuits to *exactly* obey the multiphase clocking requirements. Be sure to pay particularly close attention to the allowable clock overlap and underlap, minimum spacing, allowable width variations, the states (if any) you are allowed to stop in, and so on.

A three phase power generator circuit is shown in Fig. 25(c). It starts with a 6X clock (360 Hertz in the case of a 60 Hertz power system) and gives us three square waves phase shifted by 120 degrees. The AND gate is a disallowed state eliminator but otherwise doesn't enter the circuit.

Phase Detectors

We can also use a 4013 and an RC filter network to produce an analog output proportional to the phase shift between two signals such as shown in Fig. 26.

We use phase two or Ø2 as our reference. It sets the flip-flop with its positive edge. The positive edge of phase one (Ø1) resets the flip-flop. The time the Q output is high depends on the phase difference between Ø1 and Ø2. The output RC filter averages the high to low time ratio into a continuous analog output voltage. As the response curve shows us, the greater the phase lag of Ø1, the more output voltage you get. The best operating point is at the half-way up point at 180 degrees, as the dot in the response indicates. Operating near Ø or 360° isn't recommended due to the discontinuity as the detector slips cycles.

Sometimes we can cheat a little and make a circuit do much more than we would first suspect. Fig. 26(b) is an example of a phase detector that only works over the first 180 degrees of phase shift and only provides one-half the output voltage of the first circuit. Its best operating point is at a 90 degree phase shift. This is not as desirable as the Fig. 26(a) circuit, but probably still useful, particularly as the line lock for a TV typewriter.

But, now we can shove a plain old power line sinewave reference into the direct reset input without any conditioning, and let the *same* flip-flop do our power line conditioning and our phase detecting simultaneously. This example of do-more-with-less shows how some extra design time and rethinking can cut the amount of circuitry apparently needed in half.

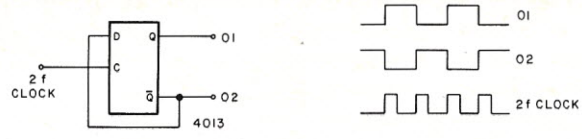
Digital Mixer

For some uses, a 4013 can act as a mixer that gives us the difference between two input frequencies. The circuit of Fig. 27 shows us the basic idea.

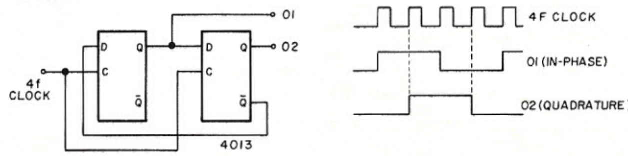
Suppose F1 and F2 were identical in frequency. The output would always be in one state. Which state depends on whether F1 was high or low at the positive edge of F2. Now, suppose that F1 is slightly lower in frequency than F1 and it starts "slipping cycles" with respect to F2. As it slips cycles, the Q output will also slip, and give us a square wave nearly equal to the difference frequency between the two inputs.

There are several restrictions to mixers of this type. Unlike analog mixers, both F1 and F2 must be single frequency, clean, digital waveforms. Our digital difference frequency is always *quantized* to be some exact submultiple of F2, so some apparent frequency jitter noise will *always* appear on the output. This jitter becomes less pronounced when you are measuring the small difference between two nearly equal high frequency signals. (Actually, this frequency jitter represents the sum term of the two frequencies. You normally can't separate sum from difference frequencies in a single mixer; it takes sine and cosine channels, single sideband techniques, or something similar.)

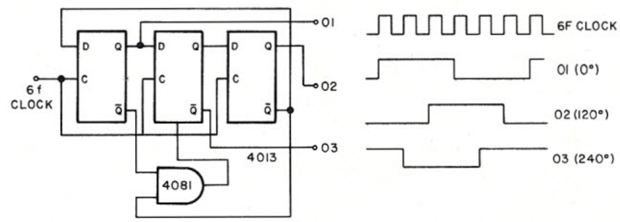
The circuit is also harmonic sensitive. For



(a) Two clock phases spaced 180°, 2f input.

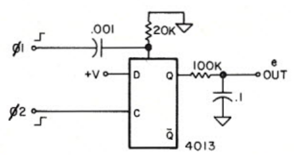


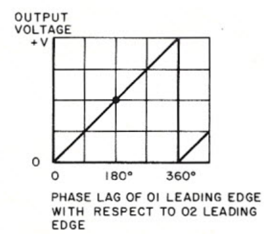
(b) Two clock phases spaced 90°, 4f input.



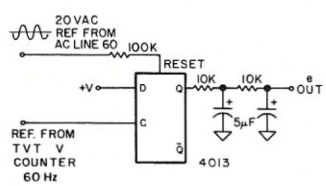
(c) Three clock phases spaced 120°, 6f input.

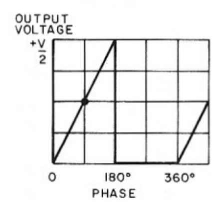
Fig. 25. Digital phase shifters form multiphase clock sources.





(a) Basic circuit.





(b) Line lock phase detector for a TV typewriter (TVT) simultaneously conditions power line reference.

Fig. 26. Clocked logic phase detectors.

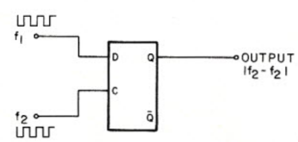


Fig. 27. Digital mixer provides difference frequency between f1 and f2.

instance, if F2 is 10 kHz and F1 is 20.003 kHz, a 3 Hertz beat note results. Mixers like this can be used in frequency synthesizers and phase lock loops. They are pretty much limited to uses where a somewhat noisy low frequency difference between two faster signals is useful.

A Tuning Indicator

We can wrap up our look at the clocked logic flip-flops with a rather stunning example of a do-more-with-less circuit. It uses a single 4013 to replace what, on first glance, would seem to be a bunch of MSI circuitry.

The problem and its solution appears in Fig. 28. The problem involves a digital data cassette recording system. To minimize errors, this recording system must be properly tuned. This is easily done by adjusting a monostable, but this could take a scope or other special test setup to do. It would be far better to come up with a "turn the knob till the red light goes out" circuit.

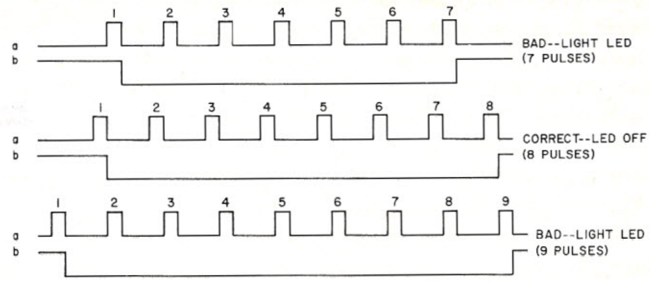
The recording system involves two signals A and B. The system is in tune if there are exactly 8 A pulses for one B pulse. If there are 7 or 9 A pulses, it is out of tune. What we want to do is build a simple and cheap circuit to put a light ON for 7 or 9 A pulses and OFF for 8.

Well, let's see. We'll need a divide-by-nine counter, a digital comparator, gating for the greater than and less than outputs, some control logic, a latch . . . a lot of parts. Can we do the whole job with a single 4013?

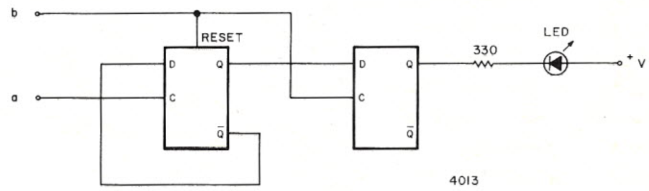
Instead of actually measuring the total number of A pulses, we observe that an *even* number (8) is valid, while the *odd* numbers 7 and 9 are not. So, we'll use half a 4013 as an even-odd detector and the other half as a result memory and LED driver, following Fig. 28(b).

Waveform B high holds the first stage in the low state. When B goes low, the first stage binary starts counting A pulses, going high on odd counts and low on even counts. When B goes back high again, the even-odd answer is stored in the second flip-flop and output to light or not light the lamp. You can do the entire circuit with a 4013, a LED, and an optional resistor.

Always watch for your chance to "do more with less" in any design problem. Very often, the simpler clocked flip-flops we have examined can eliminate or simplify things enough to greatly reduce your circuit cost and complexity. ■



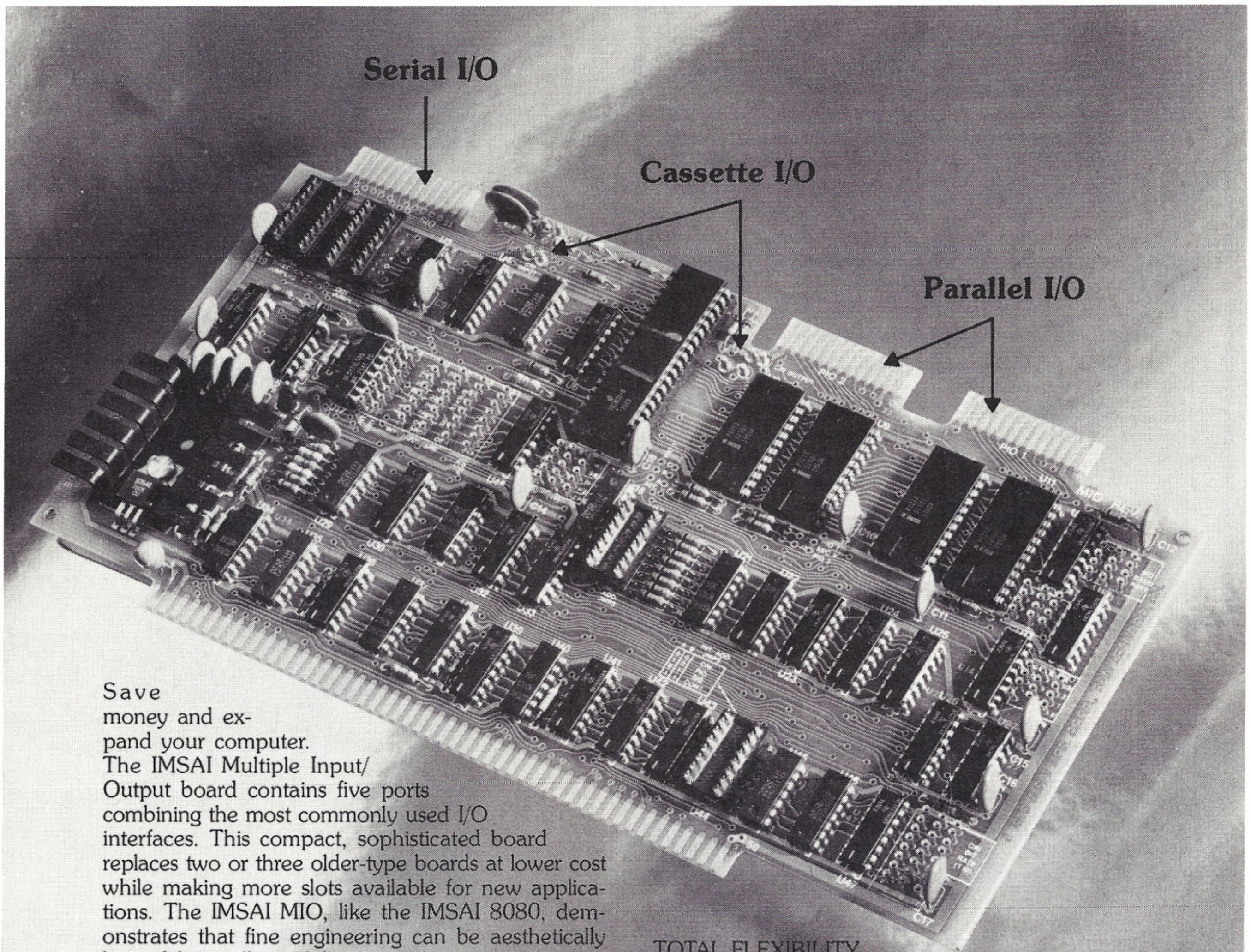
(a) The problem — show correct tuning of a bit boffer cassette interface.



(b) Solution using a single 4013. First stage is even-odd detector; second stores result.

Fig. 28. Tuning indicator for a digital cassette storage system.

IMSAI Presents:
The 3 for 1 Slot Saver
A Masterpiece in I/O Engineering.



Save money and expand your computer. The IMSAI Multiple Input/Output board contains five ports combining the most commonly used I/O interfaces. This compact, sophisticated board replaces two or three older-type boards at lower cost while making more slots available for new applications. The IMSAI MIO, like the IMSAI 8080, demonstrates that fine engineering can be aesthetically beautiful as well as solidly practical.

USEFUL AND EASY TO USE.

The IMSAI MIO is inexpensive and easy to install. It's the most advanced I/O board available, adding quality, convenience and value to your computer. With the IMSAI MIO you can control a keyboard, printer, 2 tape cassettes and a teletype/CRT simultaneously.

Consider these important "useability" features:

1. Easy to assemble components — all board identities are silk-screened.
2. Safe soldering — the entire board is solder masked for quick, fast assembly.
3. Fast, flexible configuration — options are selected with solderless wire strapping.
4. Easy testing and tuning — pre-recorded test routines on tape cassette.
5. Best of all — everything is explained in comprehensive, understandable documentation.

TOTAL FLEXIBILITY.

A control port, serial channel, 2 parallel I/O ports and a two-tape cassette interface provide all the versatility you need in an I/O board. The IMSAI MIO's flexible addressing and control features assure compatibility with non-IMSAI 8080 software. Tape cassette applications are enhanced through inclusion of the Byte/Lancaster and Tarbell recording modes.

ORDER NOW—SHIPPING NOW

The IMSAI MIO is available for immediate delivery. $195 in kit; $350 assembled. Write or call for more information. Specifications and feature product bulletin available on request. Send $1 for complete catalog of IMSAI products.

Prices: USA Domestic. Subject to change without notice.

IMSAI

IMSAI Manufacturing Corporation
14860 Wicks Blvd.
San Leandro, CA 94577
(415) 483-2093
TWX 910 366-7287

computer enterprises

Your Mail Order Computer Shop...

IMSAI 8080 kit *with 22 slots* (limited quantity)	$599.00
TDL Z-80 ZPU (the one with full software available *now*)	242.00
Edge Connectors *and* guides for IMSAI each	4.25
Edge Connectors *and* guides for IMSAI 10 for	40.00
Seals 8k RAM kit with 500 ns chips	225.00
Seals 8k RAM kit with 250 ns chips	260.00
North Star complete Micro-Disk System kit	599.00

WE TAKE
MASTER CHARGE OR BANKAMERICARD
For phone and mail orders...
(Add 4% of TOTAL ORDER for service charge)

master charge — BANKAMERICARD welcome here

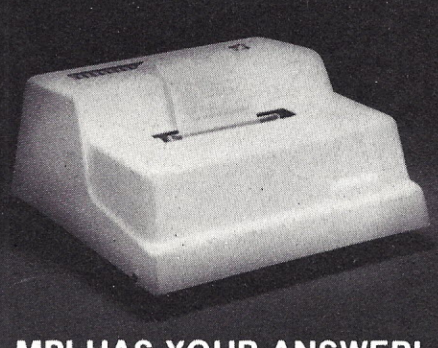
TERMS: Shipping charges — $10. per CPU or large units, $1.50 per kit, $2. minimum per order.
Provided stock is available, we will ship immediately for payment by cashiers check or money order.
Allow 3 weeks for personal checks to clear. New York State residents add appropriate sales tax.
PRICES SUBJECT TO CHANGE WITHOUT NOTICE.

For the best prices available on:

IMSAI • TDL • NORTH STAR • POLYMORPHIC
NATIONAL MULTIPLEX • SEALS ELECTRONICS

CALL: (315) 637-6208 C33
WRITE: P.O. Box 71 • Fayetteville, N.Y. 13066

Confused About Printers?



MPI HAS YOUR ANSWER!

TTY REPLACEMENT? .... THE SSP-40 **$575**
The SSP-40 contains its own microprocessor for easy connection to your serial port.

LOW COST BUSINESS SYSTEM? .... THE MP-40 **$425**
The MP-40 connects to your parallel port for ASC11 data transfer.

MINIMUM COST FOR HOBBYIST? ... THE KP-40 KIT **$179**
The KP-40 KIT contains mechanism and minimum electronics for connection to your parallel port.

All of our 40 series printers use the same reliable 5x7 impact dot matrix mechanism with up to 40 columns per line on ordinary paper with a print speed of 75 lines/minute.

MASTER CHARGE WELCOME • UTAH RESIDENTS ADD 5% SALES TAX

mpi
SEND FOR FREE LITERATURE
MICROPROCESSOR SYSTEMS AND PERIPHERALS
P.O. BOX 22101/SALT LAKE CITY/UT. 84122
(801) 566-0201

The Compucolor 8001 Is Also Available Through The Following Authorized Distributors

Phoenix Byte Shop West
12654 North 28th Drive
Phoenix, Arizona 85029
Alan P. Hald
(602) 942-7300

Tempe Byte Shop East
813 N. Scottsdale Rd.
Tempe, Arizona 85282
Alan P. Hald
(602) 894-1129

Amco Electronics
414 South Bascom Ave.
San Jose, Ca. 95128
Daniel Judd
(408) 998-2828

Computer Components
5848 Sepulveda Blvd.
Van Nuys, Ca. 91411
Dick Dickinson
(213) 786-7411

The Computer Store
63 South Main Street
Windsor Locks, Conn. 06096
George Gilpatrick
(203) 627-0188

Sunny Computer Stores, Inc.
University Shopping Center
1238A S. Dixie Highway
Coral Gables, Fla. 33146
Bill Miller
(305) 661-6042

MicroComputer Systems, Inc.
144 So. Dale Mabry Highway
Tampa, Fla. 33609
Forrest K. Hurst
(813) 879-4301

Atlanta Computer Mart
5091-B Buford Highway
Atlanta, Ga. 30340
Jim Oxford
(404) 455-0647

The Computer Mart of New Jersey
501 Route 27
Iselin, N.J. 08830
Larry Stein
(201) 283-0600

Byte Shop
2018 Greene St.
Columbia, S.C. 29205
Nick Johnson
(803) 771-7824

The Communications Center
7231 Fondren
Houston, Texas 77036
Bill Tatroe
(713) 774-9526

The Micro Store
634 S. Central Expressway
Richardson, Texas 75080
David Wilson
(214) 231-1096

Or Contact Us Direct

5965 Peachtree Corners East
Norcross, Georgia 30071
Telephone (404) 449-5961

I-6

The Compucolor 8001 System.

It's A Stand Alone Micro Computer With Color Input/Output Capabilities All In One Package. For Only $2995*.

We gave it a memory of its own.

And Floppy Tape Memory is just for starters. Look at these other features. BASIC Language, 8080 CPU, 8 color CRT Terminal, 8K RAM Workspace, Selectable Baud Rate to 9600, Two RS 232 I/O's, Keyboard, Background Color, Lower Case ASCII Characters, Roll, Insert/Delete, 48 Line X 80 Characters/Line, 2X Character Height, thorough operating instructions and a Graphics Mode with 160 X 192 Elements. And our unique Nine Sector Convergence System guarantees you quick set-up, exceptional stability and outstanding color registration in three to five minutes. If you can find a better buy in a color Intelligent CRT and Micro Computer system, let us know. We think we've got the best of both worlds at the best price going. And we want to prove it to you.

Name your game.

After all, you'll have your very own personal computer right at your fingertips. For the most simple or complex tasks. Or just plain fun. The applications are unlimited. Color graphics and computations, check book balancing, educational instruction, tutoring and a unique variety of computer games. Like Star Trek and Hangman and Pong. You can even sit back and enjoy a game of chess. Like we said, the applications are unlimited.

How about a little demonstration?

You'll find a list of our distributors on the facing page. So drop by and ask for a demonstration. Get some answers to your questions. And if you aren't near one of our distributors, give us a call. We've got the answers. The Compucolor 8001. You won't find a better buy in a color CRT Terminal and Micro Computer.

Compucolor Corporation

5965 Peachtree Corners East
Norcross, Georgia 30071
Telephone (404) 449-5961

*Unit shown includes light pen and expanded keyboard, both optional at extra cost.

Lee Wilkinson
2308 New Walland Hwy.
Maryville TN 37801

Cure Those End-of-Month Blues

... with a sales analysis program

If you don't have a small business that can benefit from this program then at least get it into your library so you can provide a good demonstration the next time you show off your system (something besides a game, right?). — John.

Most small businesses recognize a need for a sales analysis report at the end of a period of time in order to help insure their continued success. This sales analysis report gives a visual record of the categories that generate the total gross for that period of time (see Fig. 1). Column 1 indicates the categories from which you show income in your business. Columns 2 and 3 are your total sales and the number of sales in each of the categories respectively (data that you input). The fourth column reveals the average dollar sale per customer in a particular category. The fifth column gives you the percent that category contributed to the total gross sales for the period of time you wish to check. In our business we use the calendar month interval. The use of this report allows us to make decisions on when to advertise and promote in those areas which are not generating the sales we expect. Computations are simple and certainly do not necessitate a computer for the calculations. However, the computer does the calculations quickly, prints out the data in nice form and stores the data for future retrieval and examination.

This program, written in MITS 8K BASIC, is stored without data statements as the first program on the tape designated "sales analysis." The program is loaded each month and the data statements are typed in for the month or period of time you use. Where possible in my programs, as in this one, I have reserved data statements from line 1000 and up. To standardize on line 1000 for data helps me to remember

LINE NUMBER	CHANGE
15	DIM A(number of categories)
15	B(number of categories)
25	FOR I=1 TO (number of categories)
55	ON I GOSUB (line numbers of categories)
1000	DATA (date of report)
1010	DATA ($sales, no. sales)
1020	DATA ($sales, no. sales)
1030	DATA ($sales - - same month, last year)

Fig. 2. Program modification chart.

where the data statements which need to be changed reside, when I am switching from one program to another. Line 1000 contains the month and year for which you wish the printout. Lines 1010 and 1020 contain the information for computation. First is the dollar gross for the first category and second is the number of sales in the first category. All categories' information is typed in this manner. Line 1030 contains the total dollar sales from the same month of the previous year. This data is taken from our regular income summary kept by conventional bookkeeping methods.

You, of course, will change the categories to suit your own need based on your sales categories. You may expand, or decrease, the categories with slight modification to the example listing (see Fig. 2). The dimension statements in line 15 will need to be changed to the number of categories you use. In this example, A and B have been dimensioned to the 11 categories that we use. Line 25 and 50 will be changed to the maximum number of categories. Line 55 would, also, need to be adjusted to GOSUB to the first line of each category of computations. If you use fewer categories, be sure to delete the GOSUB line references on line 55. Use the same computation form as in the 11 sample categories. The FOR statement in line 350 would need to be changed to the maximum number of categories as in lines 25 and 50.

If you are using your computer for your daily income summary, this program should work nicely as an additional subroutine. An effort was made to format the program run to fit on an 8½ x 11 sheet of paper in the event additional copies are needed. Punched for a 3-ring binder, the sales analysis program gives you an instant reference when asked, "How were your sales this month?" ■

RUN

MONTHLY ANALYSIS OF WILKINSON STUDIO SALES
FOR THE MONTH OF DECEMBER 1976

TOTAL GROSS FOR THIS MONTH IS $ 4578.55
TOTAL NO. OF SALES THIS MONTH IS 46

SUMMARY OF SALES:

CATEGORY	SALES TOTAL	NUMBER OF SALES	AVERAGE PER SALE	% OF TOTAL
1 CHILDREN	693.6	8	86.7	15.15
2 STAR CLUB	0	0		0
3 COM & LGL	62.8	1	62.8	1.37
4 GROUP PORT	2001.65	6	333.608	43.72
5 ADULT PORT	895.05	3	298.35	19.55
6 SPEC PROMO	0	0		0
7 RES & CPY	483.5	16	30.2188	10.56
8 ENG & BRDL	24	2	12	.52
9 FRM & ALBM	165	9	18.3333	3.6
10 WEDDINGS	240	1	240	5.24
11 CHUR DIR	12.95	0		.28

THE AVERAGE SALE FOR THE MONTH IS $ 99.5337 PER CUSTOMER
YOU HAVE INCREASED $ 553 BY 14 %

Fig. 1. Sample run of program.

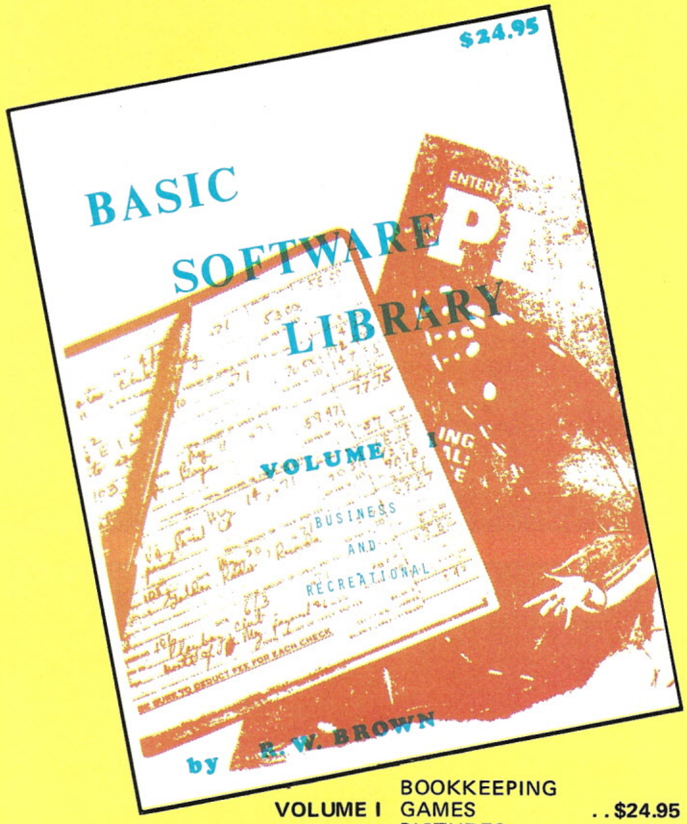
```
1 REM        ***  SALES ANALYSIS  ***
2 REM        MONTHLY DATA AT 1000 AND UP
10 FOR I=1 TO 5:PRINT:NEXT I
15 DIM A(11),B(11):T1=0:T2=0:X=0:Y=0
20 READ A$
25 FOR I=1 TO 11
30 READ A(I),B(I)
35 T1=T1+A(I):T2=T2+B(I)
40 NEXT I
45 GOSUB 70
50 FOR I=1 TO 11
55 ON I GOSUB 110,115,120,125,130,135,140,145,150,155,160
60 NEXT I
65 READ N:PRINT:GOTO 300
70 PRINT TAB(12)"MONTHLY ANALYSIS OF WILKINSON STUDIO SALES"
75 PRINT TAB(16)"FOR THE MONTH OF   ";A$:PRINT:PRINT
80 PRINT "TOTAL GROSS FOR THIS MONTH IS $";T1
85 PRINT "TOTAL NO. OF SALES THIS MONTH IS";T2
90 PRINT:PRINT "SUMMARY OF SALES:"
95 PRINT:PRINT "CATEGORY","SALES","NUMBER","AVERAGE","% OF"
100 PRINT "","TOTAL","OF SALES","PER SALE","TOTAL":PRINT
105 RETURN
110 PRINT "1 CHILDREN";:GOSUB 400:RETURN
115 PRINT "2 STAR CLUB";:GOSUB 400:RETURN
120 PRINT "3 COM & LGL";:GOSUB 400:RETURN
125 PRINT "4 GROUP PORT";:GOSUB 400:RETURN
130 PRINT "5 ADULT PORT";:GOSUB 400:RETURN
135 PRINT "6 SPEC PROMO";:GOSUB 400:RETURN
140 PRINT "7 RES & CPY";:GOSUB 400:RETURN
145 PRINT "8 ENG & BRDL";:GOSUB 400:RETURN
150 PRINT "9 FRM & ALBM";:GOSUB 400:RETURN
155 PRINT "10 WEDDINGS";:GOSUB 400:RETURN
160 PRINT "11 CHUR DIR";:GOSUB 400:RETURN
300 PRINT "THE AVERAGE SALE FOR THE MONTH IS $";T1/T2;"PER CUSTOMER"
310 Y=INT((T1-N)/N*100+.5)
320 X=INT((T1-N)*100+.5)/100
330 IF X<0 THEN PRINT "YOU HAVE DECREASED $";X;"BY";Y;"%": GOTO 350
340 PRINT "YOU HAVE INCREASED $";X;"BY";Y;"%"
350 FOR I=1 TO 11:PRINT:NEXT I
360 END
400 PRINT ,A(I),B(I),:IF B(I)=0 THEN 420;
410 PRINT A(I)/B(I);
420 PRINT ,INT((A(I)/T1)*10000+.5)/100:PRINT
430 RETURN
998 REM  DATA FORM: FIRST -- MONTH - YEAR;  NEXT -- $ SALES , NO. SALES
999 REM  LAST -- SALES FOR SAME MONTH - LAST YEAR
1000 DATA DECEMBER 1976
1010 DATA 693.60,8,0,0,62.80,1,2001.65,6,895.05,3,0,0,483.50,16
1020 DATA 24,2,165,9,240,1,12.95,0
1030 DATA 4025.55
1040 END
```

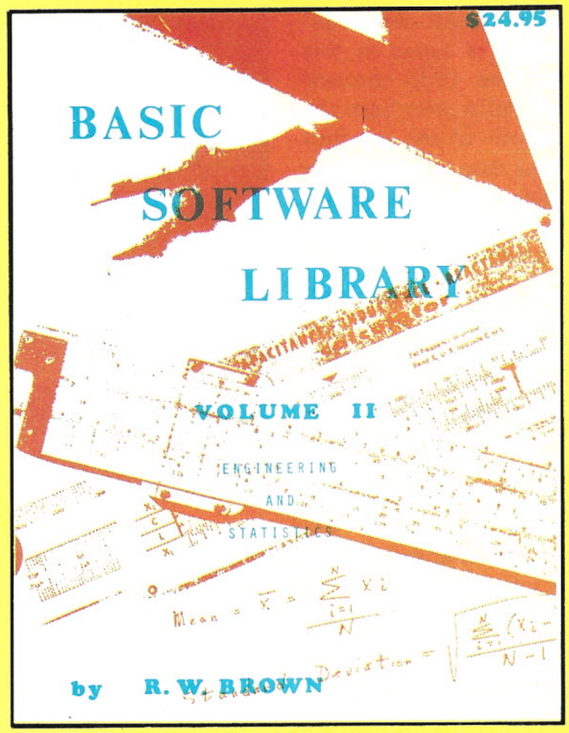
Sales analysis program listing.

The BASIC SOFT

The "HOW-TO-DO IT" Books for the "DO-IT" Person

Written in compatible BASIC immediately executable in ANY computer with at least 4K, NO other peripherals needed.





VOLUME I BOOKKEEPING / GAMES / PICTURES . . **$24.95**

VOLUME II MATH & ENGINEERING / PLOTTING & STAT / BASIC STATEMENT DEF . . **$24.95**

ARIZONA
Byte Shop of Phoenix
Tempe, AZ 85281
(602) 894-1129

Byte Shop of Phoenix-West
Phoenix, AZ 80529
/602) 942-7300

Desert Data Computer Store
Tucson, AZ 85702

CALIFORNIA
A-VID Electronics
Long Beach, CA 90806
(213) 426-5526

Byte Shop of Berkeley
Berkeley, CA 94703
(415) 845-6366

Byte Shop of Campbell
San Jose, CA 95124
(408) 377-4685

Byte Shop of Diablo Valley
Walnut Creek, CA 94596
(415) 933-6252

Byte Shop of Fresno
Fresno, CA 93703

Byte Shop of Hayward
Hayward, CA 94541
(415) 537-BYTE

Byte Shop of Lawndale
Lawndale, CA 90260
(213) 371-2421

Byte Shop of Mt. View
Mt. View, CA 94040
(415) 969-5464

Byte Shop of Palo Alto
Palo Alto, CA 94306
(415) 327-8080

Byte Shop of Pasadena
Pasadena, CA 91101
(213) 684-3311

Byte Shop of Sacramento
Citrus Heights, CA 95610
(916) 726-2557

Byte Shop of San Diego
San Diego, CA 92111
(714) 565-8008

Byte Shop of the San Fernando Valley
Tarzana, CA 93156
(213) 343-3919

Byte Shop of San Jose
San Jose, CA 95123
(408) 226-8383

Byte Shop of San Mateo
San Mateo, CA 94403
(415) 341-4200

Byte Shop of San Rafael
San Rafael, CA 94901
(415) 457-9311

Byte Shop of Santa Barbara
Santa Barbara, CA 93101
(805) 966-2557

Byte Shop of Santa Clara
Santa Clara, CA 95051
(408) 249-4221

Byte Shop/Thousand Oaks
Thousand Oaks, CA 91360
(805) 497-9595

Byte Shop of Westminster
Westminster, CA 92683
(714) 894-9131

Byte Shops, Inc.
Sunnyvale, CA 94086
(408) 734-9000

Computer Components
Van Nuys, CA 91411
(213) 786-7411

Computer Store of San Francisco
San Francisco, CA 94103
(415) 431-0640

The Computer Mart
Orange, CA 92667
(714) 633-1222

The Computer Shack
San Leandro, CA 94577
(415) 895-9363

Upland Computer Labs
Upland, CA 91786
(714) 981-1503

The Computer Store
Santa Monica, CA 90401
(213) 451-0713

People's Computer Shop
Sherman Oaks, CA 91423
(213) 789-7514

CANADA
Byte Shop of Vancouver
Vancouver 9, B.C.
(604) 736-7221

The Pacific Computer Store
Vancouver, B.C. V5R 2J4
(604) 438-DATA

Trintronics
Toronto, Ontario
(416) 598-0262

COLORADO
Byte Shop/Arapahoe Co.
Englewood, CO 80110
(303) 761-6232

Byte Shop of Boulder
Boulder, CO 80301
(303) 449-6233

The Computer Hut
Denver, CO 80202
(303) 422-7040

FLORIDA
Byte Shop of Cocoa Bch.
Cocoa Beach, FL 32931
(305) 784-1881

Byte Shop of Miami
Miami, FL 33155
(305) 264-BYTE

Computer Hut
Miami Lakes, FL 33014
(305) 821-2667

MicroComputer Systems Inc.
Tampa, FL 33609
(813) 879-4301

Micro Computer Systems & Sales
Pompano Beach, FL 33068
(305) 972-6093

Williams Radio & TV, Inc.
Jacksonville, FL 32206
(904) 354-5460

ILLINOIS
American Microprocessors Equipment & Supply Corp.
Prairie View, IL 60069
(312) 634-0076

The Data Domain
Evanston, IL 60201
(312) 328-6800

INDIANA
The Data Domain
Bloomington, IN 47401
(812) 334-3607

JAPAN
Byte Shop of Tokyo
2-9-9 Sotokanda
Chiyodaku, Tokyo
Kiyotake Ikeda

The Home Computer Store
Indianapolis, IN 46229
(317) 894-3319

KENTUCKY
The Data Domain
Lexington, KY 40502
(606) 233-3346

MARYLAND
Computer Workshop
Rockville, MD 20852
(301) 468-0455

Science Education Ext. Corp.
Potomac, MD 20854
(301) 299-9506

MASSACHUSETTS
Computer Mart, Inc.
Waltham, MA 02154
(617) 899-4540

Computer Warehouse
Boston, MA 02215
(617) 261-2701

MINNESOTA
Byte Shop of Eagan
Eagan, MN 55121
(612) 452-1841

MISSOURI
Computer Systems Center of St. Louis, Inc.
Chesterfield, MO 63017
(314) 576-5020

Computer Workshop
Kansas City, MO 64152
(816) 741-5055

NEW HAMPSHIRE
Computer Mart of NH
Nashua, NH 03060
(603) 883-2386

Microcomputers, Inc.
Nashua, NH 03060
(603) 889-1646

NEW JERSEY
Hoboken Computer Works
Hoboken, NJ 07030
(201) 420-1644

NEW YORK
Byte Shop of Levittown
Levittown, NY 11756
(516) 731-8116

Computer Mart of Long I
East Meadow, NY 11554
(516) 794-0510

Mini Micro Mart
Syracuse, NY 13202
(315) 422-4467

OHIO
Digital Design
Cincinnati, OH 45243
(513) 561-6733

The Computer Mart of NY
New York, NY 10001
(212) 279-1048

Synchro-Sound Enterprises
Hollis, NY 11423
(212) 468-7067

OKLAHOMA
High Technology
Oklahoma City, OK 73116
(405) 842-2021

OREGON
Byte Shop of Beaverton
Beaverton, OR 97005
(503) 644-2686

Byte Shop of Portland
Portland, OR 97201
(503) 223-3496

PENNSYLVANIA
Byte Shop of Bryn Mawr
Bryn Mawr, PA 19010
(215) 525-7712

Personal Computer Corp.
Frazer, PA 19355
(215) 647-8460

RHODE ISLAND
Computer Power Inc.
Warwick, RI 02886
(401) 738-4477

SOUTH CAROLINA
Byte Shop of Columbia
Columbia, SC 29205
(803) 771-7824

TENNESSEE
Byte' Tronics
Nashville, TN 37203
(615) 329-1979

Micro Computer Systems
Knoxville, TN 37922
(615) 966-9849

TEXAS
Altair Computer Center
Houston, TX 77036
(713) 780-8981

Interactive Computers
Houston, TX 77036
(713) 772-5257

Southwest Technical
San Antonio, TX

UTAH
Byte Shop/Salt Lake City
Salt Lake City, UT 84111

VIRGINIA
Computer Hobbies Un
Richmond, VA 23235
(804) 276-5056

Media Reactions Inc.
Reston, VA 22090
(703) 471-9330

Shire Enterprises
Richmond, VA 23222
(804) 321-4560

WASHINGTON
Almac/Strom Electronics
Seattle, WA 98108
(206) 763-2300

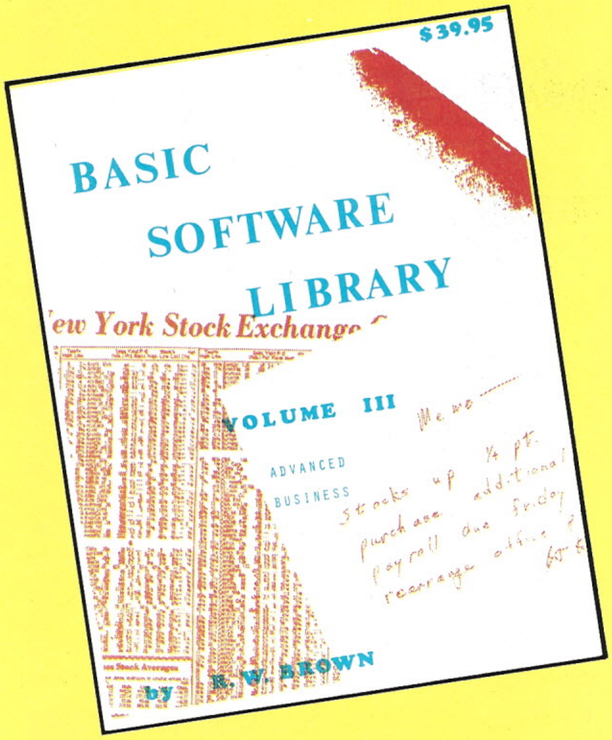
WISCONSIN
Microcomp
Fond du Lac, WI 54935
(414) 922-2515

The Milwaukee Computer Store
Milwaukee, WI 53213
(414) 259-9140

STOP BY OUR BOOTH (e615) AT THE SAN

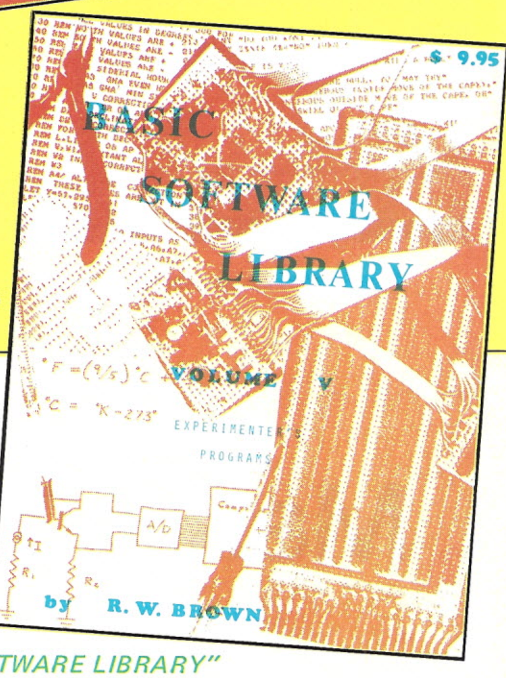
WARE LIBRARY

This LIBRARY is a complete do it yourself kit.
Knowledge of programming not required.
EASY to read and USE.



VOLUME III ADVANCED BUSINESS
BILLING
INVENTORY ..$39.95
PAYROLL

VOLUME IV
GENERAL PURPOSE ..$9.95



VOLUME V
EXPERIMENTER ..$9.95

This library is the most comprehensive work of its kind to date. There are other software books on the market but they are dedicated to computer games. The intention of this work is to allow the average individual the capability to easily perform useful and productive tasks with a computer. All of the program contained within this Library have been thoroughly tested and executed on several systems. Included with each program is a description of the program, a list of potential users, instructions for execution and possible limitations that may arise when running it on various systems. Listed in the limitation section is the amount of memory that is required to store and execute the program.

Each program's source code is listed in full detail. These source code listings are not reduced in size but are shown full size for increased readability. Almost every program is self instructing and prompts the user with all required running data. Immediately following the source code listing for most of the programs is a sample executed run of the program.

The entire Library is 1100 pages long, chocked full of program source code, instructions, conversions, memory requirements, examples and much more. ALL are written in compatible BASIC executable in 4K MITS, SPHERE, IMS, SWTPC, PDP, etc. BASIC compilers available for 8080 and 6800 under $10 elsewhere.

This Library is destined to become one of the reference bibles for the small computer field, due to its versatility and uniqueness and the ease of operation of the programs it contains. These volumes are deductible as a business expense when purchased by a company. Send your remittance for prompt delivery, while supplies last. Volume discounts are available to qualified dealers.

FUTURE ADDITION TO THE "BASIC SOFTWARE LIBRARY"
Volume VI (A Complete Business System — $49.95) General Ledger System — Taxes, Pyrl, W-2's, Inventory, Depr., Financial Statements, etc. AVAILABLE MID SUMMER

SCIENTIFIC RESEARCH

1712-K FARMINGTON COURT
CROFTON MD 21114
Phone Orders call (800) 638-9194
Information and Maryland Residents Call (301)-721-1148

master charge THE INTERBANK CARD

Your BANKAMERICARD welcome here

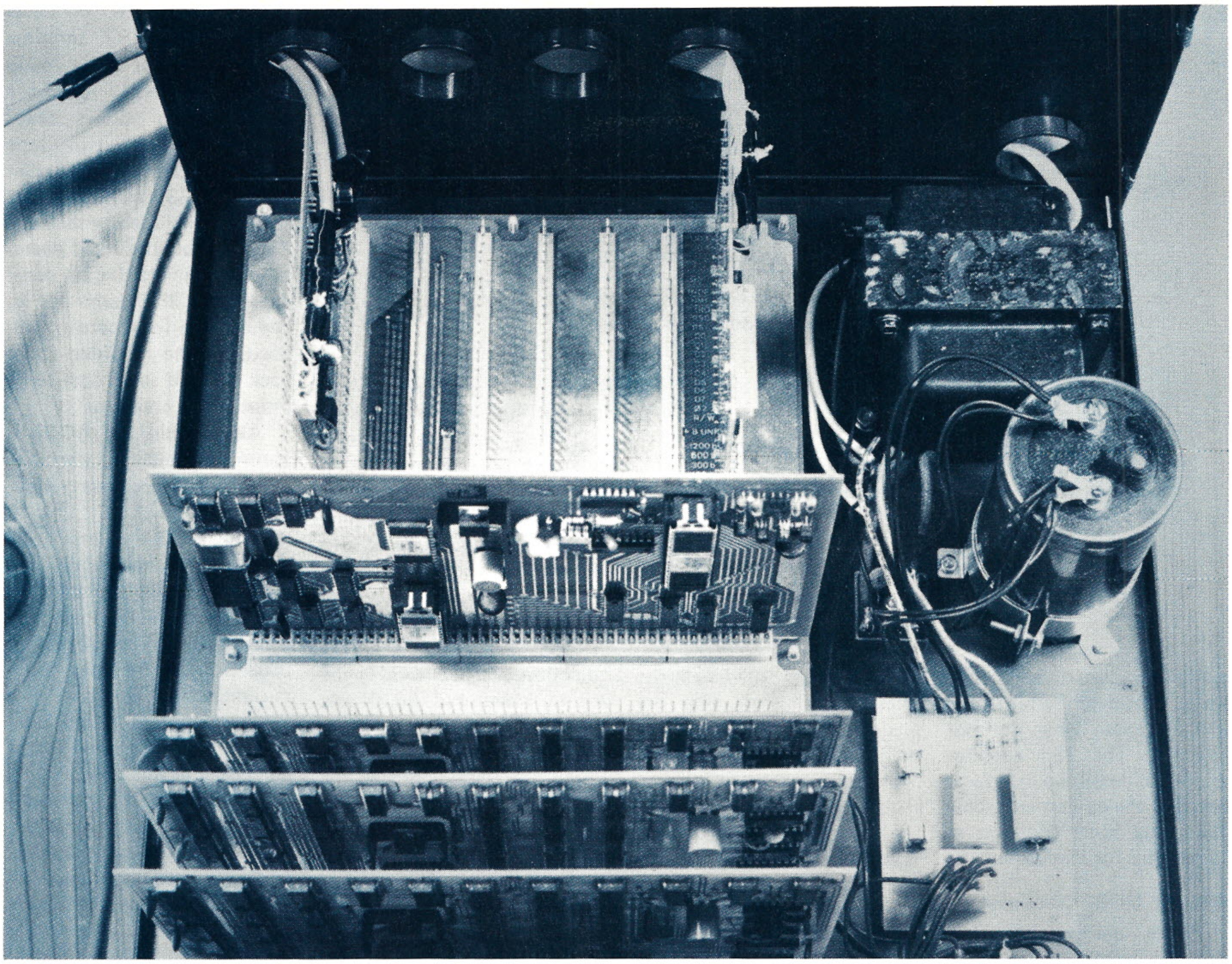
Add $1.50 per volume for postage and handling.

FRANCISCO COMPUTER FAIRE and say "HI!"

The complete system. MP-68, ADM-3K terminal, AC-30 cassette interface, PR-40 printer, 2 cassette recorders, and a library of tapes.

Make Your Investment Count

... the inside view of a custom MP-68



Inside the MP-68. This is the complete system with 12K bytes of memory.

Phil Hughes
PO Box 2847
Olympia WA 98507

In a recent editorial in 73 Magazine *Wayne Green was discussing the costs of home computer systems. His comments are worth quoting here as an introduction to Phil Hughes' article: "Why do I keep hearing you mumbling about cost? This is a* hobby *. . . like sports cars . . . flying . . . blondes . . . and, as you know, the difference between a man and a boy is the price of his toy. You can add up all the components for yourself, but it is unlikely that you are going to have much going for under $2,000." — John.*

This article describes how a beginning computer hobbyist can purchase a computer system in kit form with all the necessary memory, software, interfaces, and peripheral equipment to do *real* computing for about $2000. The proposed system consists of a processor with 12K bytes of memory, 2 cassette tape units, a cassette interface, a commercial quality CRT terminal, an inexpensive printer and sufficient software for programming extensively in both BASIC and assembly language.

As well as presenting a complete system, alternate equipment is discussed and interface details presented.

Proposed System

The proposed system is based on the Southwest Technical Products Corporation (SWTPC) MP-68 Computer System Kit. Added to this is a kit by Lear Siegler, a commercial CRT terminal manufacturer and two standard audio cassette recorders. There is probably a *better* device for each one presented, however the purpose of this article is to suggest one possible beginning system that is expandable, has sufficient support to make it easy to use and will protect as much of your initial investment as possible.

Each of these tradeoffs will be discussed individually.

If you have no kit-building experience you should locate someone with an electronics background to be available if you need help. If you have successfully built other large electronics kits you should have little or no trouble with these kits.

The major components of the system are shown in Table 1.

To complete the system approximately $40 of additional equipment and supplies will need to be purchased. Most of this should be available at a local radio or hi-fi store. It consists of items shown in Table 2.

The following sections of this article will discuss in detail each piece of equip-

No.	Manufacturer	Model	Description	Cost
1	SWTPC	MP-68	Computer w/4K memory	$ 395
2	SWTPC	MP-M	4K byte Memory	200
1	SWTPC	MP-L	Parallel Interface	35
1	SWTPC	MP-E	Editor/Assembler Package	15
1	SWTPC	AC-30	Cassette Interface	80
1	SWTPC	PR-40	Printer Kit	250
1	SWTPC	BA-8K	8K BASIC Interpreter	10
2	Sanyo	M2533	Cassette Recorder	120
1	Lear Siegler	ADM-3K	Terminal Kit	875
			Total	$1980

Table 1. Major components of the system.

4 RCA Phono to Min. Phone Plug Cables	$ 6
2 RCA Phone to Sub-Min Phone Plug Cables	3
20 feet of 12 conductor stranded cable	5
1 Cinch D-Series 25 Pin Male Connector	5
10 Cassette Tapes	20
Total	$39

Table 2. Additional equipment.

ment (hardware and software), present alternatives and discuss why it was selected. The last sections discuss in detail the interfaces between the equipment, some debugging hints and present some ideas for expansion.

Why the MP-68 Computer

There are four reasons why the MP-68 was selected. First is the microprocessor chip the system is based on. This is the Motorola 6800. Currently there are two very popular microprocessor chips in hobby computing, the 6800 and the 8080. The 6800 was developed by Motorola and the 8080 by Intel (although other companies are currently second-sourcing them). Other processor chips, such as the Fairchild F-8, National PACE, and Zilog Z-80 are becoming popular but are either much more expensive than the 6800 or have little or no software available which is of use to the hobbyist.

If we were to take a close look at the design of the instruction sets of the 6800 and 8080 we would see that the 6800 is more like a minicomputer than the 8080 with fewer *quirks* to confuse the beginner. For example, in the 8080 if an instruction uses a 16-bit address, the low order (least significant) bits are placed in the instruction before the most significant. This is equivalent to learning that the number one thousand two hundred and thirty-four is written 3412. Not really hard to learn but certainly confusing at first.

The second reason for selecting the MP-68 is because of a Motorola firmware routine (software in read-only memory) called MIKBUG*. MIKBUG is a set of routines which takes the place of a switches and lights console. Built into this firmware package is a memory display/change routine, load and dump routines, interrupt handlers and various input/output (I/O) routines. Using MIKBUG from your control terminal is much easier than storing and displaying data using console switches. Another advantage is the reduction in the manufacturing cost of the processor by the elimination of switches and lights.

The third reason for selecting the MP-68 is software. There is almost as much software available for the 6800 as there is for the 8080. Also the attitude of SWTPC is to distribute software at their cost and publish information on other available software rather than making it a high profit item. Because of this policy more 6800 software will be coming and most likely at reasonable prices.

The last reason for the selection of the MP-68 is the kit manufacturer, Southwest Technical Products Corp. SWTPC has been building audio and test equipment kits for years and knows what kits are all about. Note, however, that this is not like a Heathkit. You do not get an instruction for each part in the kit, but are more likely to see *put all the resistors on the circuit board.*

The quality of the parts in the kits is excellent. They are not surplus. Costs appear to be cut only in the elimination of extremely detailed instructions (if there is something tough to look out for, it will be detailed) and in eliminating frills like fancy cabinetwork. Also having been in the hobby kit business for years (unlike the other computer kit manufacturers), SWTPC is used to diagnosing and if necessary correcting kit building errors.

Cassette Storage

The audio cassette using the Kansas City standard is the least expensive form of mass storage available to the hobbyist. It is also a very popular storage method; therefore, it can be used for program and data interchange with other hobbyists.

The SWTPC AC-30 cassette interface connects between the CRT terminal (or any other 300 baud ASCII terminal) and the computer. It allows read/write access to two tape recorders. Data is recorded at 300 baud (30 characters per second). Included within the AC-30 is all the logic necessary to convert the bit serial data from the computer to audio tones for recording and decoding the recorded tones back into bit serial data to be input to the computer.

Also within the AC-30 is control circuitry to start and stop two recorders under program control.

The AC-30 offers compatibility with both the MP-68 and the Kansas City standard tape format and offers excellent control of the mass storage facilities for a minimum of expense. For these reasons it is my only choice for a mass storage controller at this time.

The choice of a cassette recorder is another matter. There are many excellent audio cassette recorders available and probably the biggest influence in the selection is what is available locally. I chose the Sanyo M2533 because of its features, cost and local availability. Superscope* recorders were used by SWTPC in their testing and I initially used a 6-year-old Lloyd's Model 2V96A until a mechanical problem developed which was causing damage to tapes.

Rather than specifying a particular *must get* recorder here are some features to look for:

- Push button operation – Ease of use.
- 117 V ac Operation – Batteries run down and cause speed change. Battery replacement also increases the operating cost.
- Tape counter – For locating programs.
- Remote control – Opera-

*MIKBUG is a registered trademark of Motorola, Inc.

*Superscope is a trademark of Superscope Inc.

Experience the excitement of owning the world's finest *personal* computer – IMSAI 8080

Waiting for you — all the incredible performance and power of the IMSAI 8080. And at a price you would normally pay for a fine home music system.

Introduced less than two years ago, the IMSAI 8080 is sold worldwide and acknowledged as the finest *personal* computer available.

WORLD OF USES

The IMSAI 8080 is a superbly engineered, quality computer. It is versatile, expandable and powerful, putting literally hundreds of applications and uses at your fingertips. Imagine sitting at your desk and enjoying interaction with your IMSAI 8080! Press the *on* switch and you're ready for game playing, research, education, business applications, or for laboratory instrument control. It has all the power you need, and more, to make your application ideas come alive.

GROWS WITH YOU

The IMSAI 8080 is designed for many years of pleasure. With its open-chassis engineering you can expand your system by adding peripherals and interfaces. The 22-slots and 28 amp power supply mean that you can plug-in today's, plus *tomorrow's* modules.

Right now you can add a module for displaying color graphics and characters on TV; a ready-to-use keyboard; small and large printers, and a single interface that lets you attach multiple devices including a cassette tape recorder. Expect the latest, exciting equipment from IMSAI. We are committed to leadership in this expanding technology.

EASY TO PROGRAM

With our BASIC language you can operate the IMSAI 8080 quickly and easily. Technically knowledgeable? Use our assembly language to develop sophisticated and unique software.

If you're thinking personal computer, treat yourself to the very best – IMSAI 8080

Send for free four-color brochure or $1 for catalog. Call or write for name of nearest dealer.

IMSAI

IMSAI Manufacturing Corporation
14860 Wicks Blvd.
San Leandro, CA 94577
(415) 483-2093
TWX 910-366 7287

tion can be controlled by the interface.

- Rewind operated when remote control is off – This is not mandatory but it makes the recorder easier to use. You can rewind one recorder while the other is operating under program control.

Most recorders which offer all these features will be of high enough quality to work reliably in this environment. One caution, however; purchase two recorders exactly the same. If they are different models the tape counter will probably operate differently. This happened to me when I purchased a Sanyo M2533 and a Sanyo M2522. Both recorders operate very well (although the M2522 will not rewind with the remote switch off) but the tape counters indicate differently. This means that I have to use the same recorder for playback as for record if I need to locate a program which is not at the start of a tape.

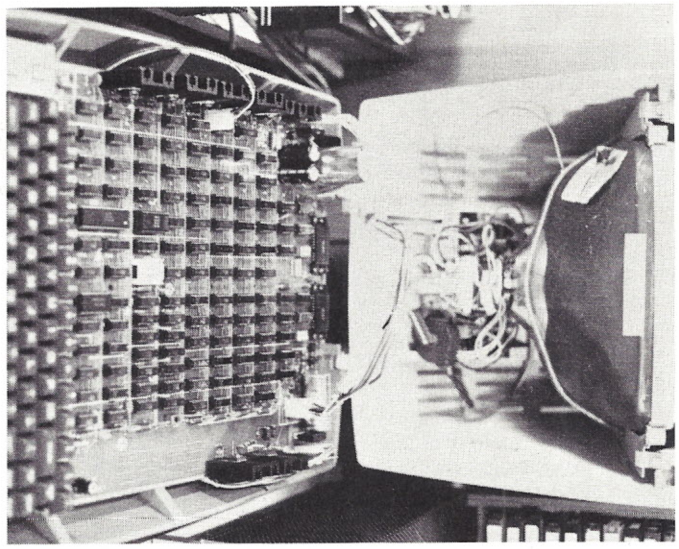
Tape selection is much like cassette recorder selection. Audio cassette tapes are another consumer product where there are a lot of good products; so just look for features, cost and availability. I have used and had no problems with the following cassette tapes:

- BASF Studio Series C-60
- Maxell UD-XL C-60
- Memorex MRX2 Oxide C-45
- Scotch Classic C-60
- TDK SD C-45

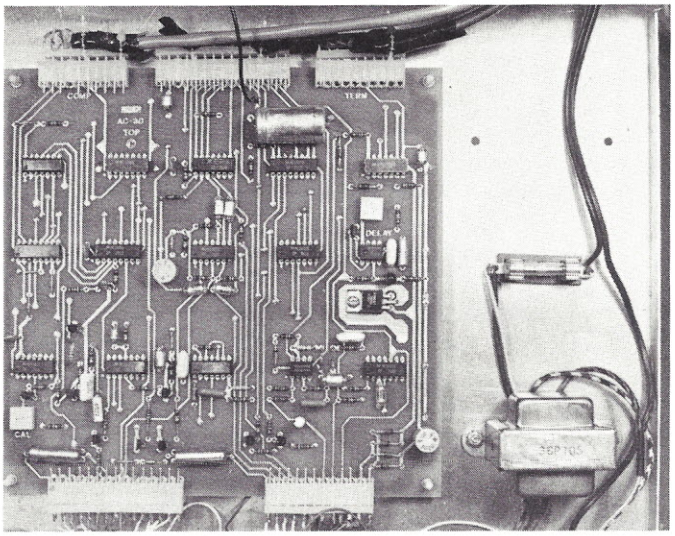
Of these tapes I have been using Memorex because it is available at lower cost than the others I tried. There are two cautions in purchasing cassette tapes. Buy good quality tapes, not because you need great fidelity, but because you need a tape free of dropouts. Purchase tapes no longer than a C-60, with a C-30 or C-45 being preferred. This is because the longer tapes are thinner and therefore are more likely to stretch or break.

The Lear Siegler Terminal

The Lear Siegler ADM-3K CRT terminal kit is a commercial unit that is now available in kit form. Lear Siegler makes 3 CRT terminals (ADM-1, -2, and -3) and sells them both directly to users and to systems manufacturers for incorporation into their product line (OEMing). The ADM-3 is the lower end of the Lear Siegler line both in price and features. For its low cost, however, it offers a large screen (12 inch diagonal) and a large display capacity (24 lines of 80 characters). This display capacity is the most popular in the industry and will also probably become the hobbyist standard when CRT terminal prices drop.



Inside the ADM-3K. It opens like a clam shell for each access.



Inside the AC-30. Only two adjustments (delay and cal) are required.

If you purchase a CRT terminal (also called a TV typewriter) such as the SWTPC CT-1024, you end up with what looks like a homemade unit. You must supply the case and a television set for the display. This will only cost about $375 ($275 for the CT-1024 with the necessary options plus about $100 for a television) but you end up with a display of only sixteen lines of 32 characters per line. Also the CT-1024 is paged, not scrolled. What this means is when you have filled up 16 lines with information you must switch to the second page of the display. In a scrolled display, such as the ADM-3, new information is always entered on the last line of the display and is rolled upward. With scrolling the last 24 lines are always visible.

The main reason I am dwelling on the CRT terminal is that, if you move up to a better system, get into using a time-sharing system, or just bail out of the computing hobby altogether, the commercial terminal will retain much more resale value than the TV typewriter kits and also will be better suited for time-sharing.

As far as the construction of the ADM-3K there is a lot of information included with the kit. The information is not presented as well as it could be, but with some digging all that is necessary for construction is there. The kit itself is laid out for easy access. The CRT and deflection circuits are all located in the top half of the case and the logic, memory and power supply are all on one large circuit board in the bottom half of the case. The case is hinged at the back and opens up like a clam shell so you can reach everything.

The PR-40 Printer

With the SWTPC PR-40 printer I have gone the opposite way as with the CRT terminal. The PR-40 is the least expensive new printing device which you can hook up to your microprocessor. It features the dot matrix printing method which has been used by Centronix and other makers of commercial slow to medium speed printers. The biggest drawback is its 40 column width. If you intend to mainly use the printer for listing programs and not typing letters, it is adequate. A printer of essentially the same speed (75 characters per second) but 80 columns in width is currently in the $1500 price range.

One other alternative is to replace the CRT terminal with a hard copy terminal. In order to be compatible with the AC-30 cassette interface this terminal must be capable of operation at 300 baud. One of the least expensive units capable of operation at this speed is the Digital Equipment Corporation LA-36 Decwriter. The Decwriter costs about $2000 (less from some teminal dealers) which is more than the cost of the ADM-3K and the PR-40.

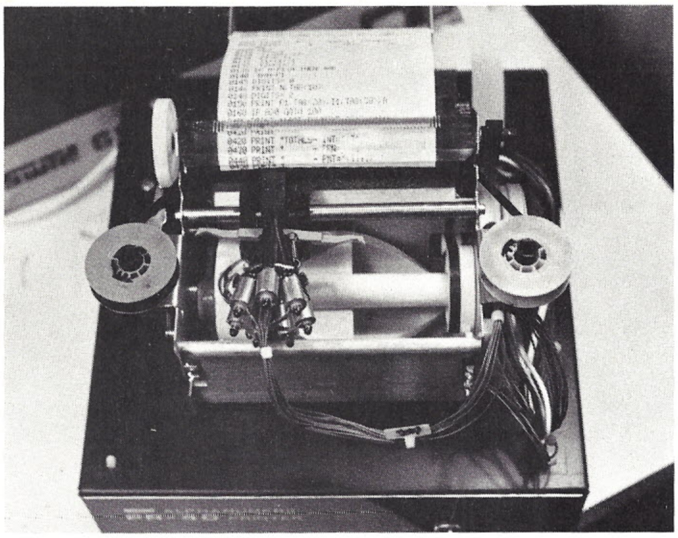
The other advantage is the PR-40 is designed to plug into a SWTPC MP-L interface board and start working. There are no software changes to make and no hardware interface to build.

Software

It has been said too many times but here it is again, *a computer is useless without software*. Software is the thing which transforms a box of electronics to a useful tool or a toy or both. Some of the manufacturers have tried to make software the expensive part of the system. SWTPC offers diagnostic software free with the system, Micro-BASIC and games like Tic-Tac-Toe and Blackjack free in a newsletter and an editor, assembler and full BASIC for nominal media and documentation charges.

SWTPC offers two BASIC interpreters, 4K BASIC and 8K BASIC. 4K BASIC costs $4.95, runs in as little as 6KB and offers everything in the BASIC language except mathematics functions such as LOG, and SIN. This BASIC will get you started but it does not include interface software to the PR-40 (although patches to accomplish this interface were published in the last SWTPC newsletter).

A better choice for the 12KB system under consideration is the 8K BASIC. This interpreter includes all the features of a full BASIC and some extras in string handling and input/output operations to other devices (including the PR-40). This is the interpreter that I have been using



The PR-40 printer. Paper is standard 3 7/8 inch adding machine rolls.

and it is relatively bug-free, which is amazing considering it was just released. I spoke to the author concerning the bug I found and one was caused by an error in tape duplication and the other (being able to save a line which is too long to input) will be corrected in the next version. Also, 8K BASIC will not start and stop the cassette recorders under program control. This change will probably be incorporated in the next version also. The only logic error I thought I found when executing the interpreter turned out to be a hard-to-catch memory bug. (This bug was my fault. I tried those cheap surplus memory ICs.)

Most of the game programs will run in 12KB when using 8K BASIC. I have a version of KING (one of the largest games) which I tailored to fit in 12K. The major change I made was to remove the instructions that were typed out at the start of the game. By typing them on a sheet of paper the player can refer to them throughout the game. He would not have been able to do this if they were displayed on the CRT at the beginning of the game. This along with a $100 saving in memory cost seems like a good justification for typing the directions.

The editor software works very well. I have found no bugs and once you get used to its commands it is versatile and easy to use. It is not line-oriented and therefore it could be used for editing data files as well as assembly language programs. The only thing missing is the ability to output text to a device other than the control port (cassette interface and CRT terminal). Robert Uiterwyk (the author of 8K BASIC) is updating the editor and assembler so I would expect the next version to interface to the PR-40.

The assembler appears to be the same one developed by Motorola with changes in the peripheral device handlers to work with the AC-30 cassette interface. Load time is quite long (about 10 minutes) but it works quite well.

Included with the system are two memory diagnostics and a diagnostic for each piece of peripheral equipment. Each diagnostic is short enough to type in from the CRT terminal and appears to do a good job of locating errors. In the case of the AC-30 and PR-40 the diagnostics are used to make adjustments. The only test equipment required is a voltmeter.

Membership in the Motorola 6800 users group is offered to purchasers of the MP-68. The cost of membership is $100 or a piece of software. Such things as multiply routines, diagnostics, and special I/O routines are found in the users group library. If you are interested in assembly language programming then trading one of your routines for membership is probably a good investment.

Interface Details

Most of the interface information is covered in the manual for the equipment being interfaced. The interface to a 300 baud terminal which is not a SWTPC CT-1024 is covered in general terms. Here are the missing specifics.

The ADM-3 must be configured for the following:

- Bit 8 – 0, Sets data bit 8 to a 0.
- Parity – INH, Inhibits

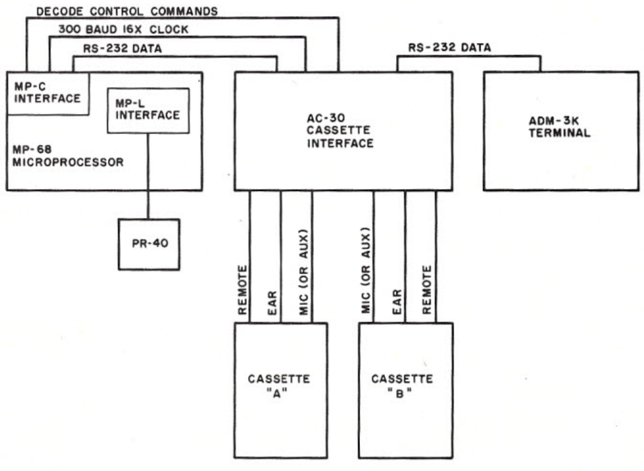


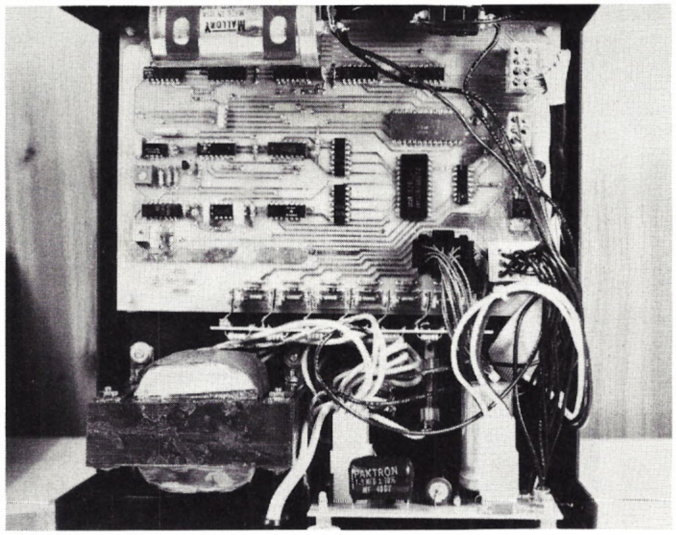
Fig. 1. System block diagram.

character parity.

- Stop — 1, One stop bit.
- Data — 8, Selects 8 data bits per character.
- LC — UC, Transmits only upper case characters.
- RS 232 — RS 232, Selects the RS-232 interface levels.
- FDX, Selects full duplex mode. Characters are echoed to the display by the computer.
- Baud rate — 300, Selects 300 baud.

Fig. 1 shows a block diagram of the complete system. The cables shown in the SWTPC 6800/CT-1024 interconnection drawing should be prepared using the 12 conductor wire. Note that not all 12 conductors are needed. The length of the two I/O cables (from the processor to the AC-30 and from the AC-30 to the terminal) can be any length up to about 250 feet. Make them long enough so you can move the equipment around without having to recable. The cable from the AC-30 control connector will be run to the MP-C interface card in the MP-68 processor. This is covered in the section in the AC-30 instructions titled "Interfacing to a SWTPC 6800 and non-CT-1024 300 baud Terminal System." Keep this cable as short as possible as it is connected directly to the Peripheral Interface Adapter and long cables could introduce noise on this signal line. The two pins which are indicated as "Terminal's 16X clock OUT," and "Terminals 16X clock IN," should not be connected at this time. These allow you to key data directly from the terminal to cassette tape. If you feel comfortable digging into the ADM-3, these can be connected; but they are not really necessary for most operations and could cause damage to the AC-30 or ADM-3 if not connected properly.

The cable for the printer should be prepared as shown on the PR-40/SWTPC 6800 Interconnection Diagram. This cable can be about any reasonable length. Make it long enough so that you can move the printer to work on it.



The interface electronics are in the base of the PR-40.

If you were unable to locate premade cables to run from the cassette recorders to the AC-30 interface, they can be made out of any shielded cable. Length of these cables is also not critical, make them long enough to move things around.

If you have everything ready you should start testing. Rather than connect the complete system together, try to test in the smallest configuration possible and then add and test each additional unit. For the proposed system the following testing sequence should minimize debugging problems:

1. Test the terminal as a stand-alone unit.
2. Connect the MP-68 with only the MP-A System Board and MP-C control interface installed to the terminal and test these units.
3. Add the AC-30 cassette interface and recorders and perform the necessary adjustments.

Note: At this point you can start saving diagnostic programs on cassette tape.

4. Add each memory board and test using both the ROBIT and MEMCON diagnostics. Run them on all the memory each time, not just the new board that you added.
5. Add the MP-L parallel interface to I/O slot 7 and test using the PARINT diagnostic.
6. Connect the PR-40 printer to the MP-L interface and perform the necessary adjustments.

At this point your complete system should be configured as shown in Fig. 1. You can now start loading the software packages and have at it. You have a complete computer system.

Expansion

If you have some more money and want your computer to do something new, equipment is available to expand it. You did not buy a dead-end system.

The first expansion item is memory. If you have found a new program or game that takes more memory than you have, you can buy another memory board. At this time the rumors say that SWTPC will be offering an 8K board as well as the 4K boards in the proposed system. This may be the way to go.

Graphic capabilities is another expansion option. Circuits and ideas for graphics have been published in almost all the hobby computer magazines. If you are not ready to attack building from scratch, SWTPC offers the GT-6144 Terminal System. This unit connects to a parallel interface (MP-L) and will display an array of cells 64 X 96 on a video monitor or modified television set. It costs $98.50 minus power supply and chassis, and should be considered as an inexpensive move into graphics. Like the PR-40, unless you build something on your own, the step up from the GT-6144 is in the vicinity of $2-10,000 for a commercial graphics terminal.

Disk storage is another expensive but desirable option. Although currently there are no disk systems that are specifically designed to interface to the SWTPC 6800 they are on their way. Also, floppy disks with programmable controllers can be interfaced without a lot of work. By the time this article is published I expect that some company will be offering a disk which interfaces to the SWTPC 6800. Also, it is rumored that SWTPC is working on a disk operating system. This interfaced floppy disk will be in the $500 (minifloppy with dumb controller) to $1500 (regular floppy with intelligent controller) range. When you are ready to buy, look around.

This covers the proposed system. If you are still looking around for an inexpensive yet viable system, then this is one of the currently available alternatives. A year from now there may be a new system which costs only $1000, runs twice as fast and has more software available for it than the 6800. I doubt that this will be the case. More likely there will be a new, better processor chip which will be pin-for-pin compatible with the 6800, will fit into your system and will use the same software. Even if the system board must be replaced to interface to the new processor, much of your investment will be protected. Happy computing! ■

The POLY 88 Microcomputer System

PolyMorphic Systems now offers the complete, assembled, personal computer system—the POLY 88 System 16. A full 16K system with high speed video display, alphanumeric keyboard, and cassette program storage. A BASIC software package providing the most advanced features available in the personal computing market. Features like PLOT and TIME, which utilize our video graphics and real-time clock. Others like VERIFY, so that you know your tape is good before you load another. Or input type-ahead so you can tell your program to run while the tape is still loading (it stores up to 64 characters of commands or question responses to be executed). All these plus a complete package of scientific functions, formatting options, and string capabilities. With the POLY 88 System 16 you can amaze your timesharing friends the very first night!

PolyMorphic Systems 11K BASIC

Size: 11K bytes.
Scientific Functions: Sine, cosine, log, exponential, square root, random number, x to the y power.
Formatted Output • Multi-line Function Definition • String Manipulation and String Functions • Real-Time Clock • Point-Plotting on Video Display • Array dimensions limited by memory • Cassette Save and Load of Names Programs • Multiple Statements per Line • Renumber • Memory Load and Store • 8080 Input and Output • If Then Else • Input type-ahead.

Commands: RUN, LIST, SCR, CLEAR, REN, CONT.

Statements: LET, IF, THEN, ELSE, FOR, NEXT, GOTO, ON, EXIT, STOP, END, REM, READ, DATA, RESTORE, INPUT, GOSUB, RETURN, PRINT, POKE, OUT.

Built in Functions: FREE, ABS, SGN, INT, LEN, CHR$, VAL, STR$, ASC, SIN, COS, RND, LOG, TIME, WAIT, EXP, SQRT, CALL, PEEK, INP, PLOT.

Systems Available. The POLY 88 is available in either kit or assembled form. It is suggested that kits be attempted only be persons familiar with digital circuitry. The following are two of the systems available.

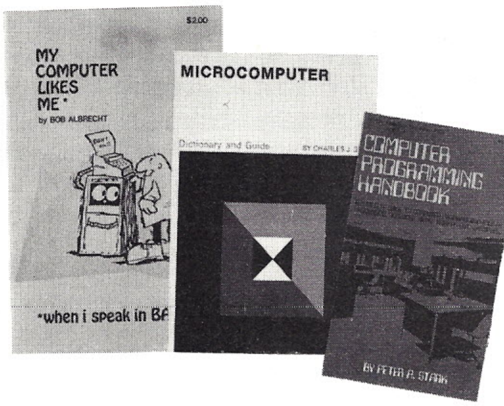
System 2: is a kit consisting of the POLY 88 chassis, CPU, video circuit card, and cassette interface. Requires keyboard, TV monitor, and cassette recorder for operation. $690

System 16: consists of an assembled and tested System 2 with 16K of memory, keyboard, TV monitor, cassette recorder, 11K BASIC and Assembler on cassette tapes. $1995.

Prices and Specifications subject to change without notice.
California residents add 6% sales tax.

PolyMorphic Systems
460 Ward Drive Santa Barbara, Ca. 93111 (805) 967-2351

KB BOOK NOOK



●**BRAND NEW DICTIONARY** This new microcomputer dictionary fills the urgent need for all computer people, engineers, scientists, industrialists, communications people — as professionals, amateurs, teachers, or students — to become quickly acquainted with the terminology and nomenclature of a new revolution in computer control capabilities in areas that pervade most of man's daily activities.

Over 5000 definitions and explanations of terms and concepts (704 pages) relating to microprocessors, microcomputers and microcontrollers. There are also separate appendices on: programmable calculators; math and statistics definitions; flowchart symbols and techniques; binary number systems and switching theory; symbol charts and tables; summaries of BASIC FORTRAN and APL. In addition there is a comprehensive electronics/computer abbreviations and acronyms section. $15.95.

●**COMPUTER PROGRAMMING HANDBOOK** by Peter Stark. A complete guide to computer programming and data processing. Includes many worked out examples and history of computers. $8.95

●**MY COMPUTER LIKES ME . . . WHEN I SPEAK BASIC** An introduction to BASIC . . . simple enough for your kids. If you want to teach BASIC to anyone quickly, this booklet is the way to go. $2.00.

●**SCELBI'S GALAXY GAME FOR THE "8008"/"8080"** Here's a new twist in computer games by Scelbi Computer Consulting and Robert Findley. The game, "Galaxy", pits the operator of a spaceship against alien craft, as well as variables such as speed, time and ammunition. No two games are the same! This game is described in *Galaxy Game for the 8008/8080*, published by Scelbi Computer Consulting, Inc. $14.95.

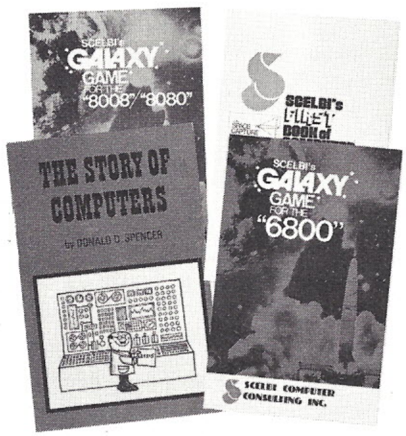
●**SCELBI'S GALAXY GAME FOR THE "6800"** Here's a new twist in computer games by Scelbi Computer Consulting and Robert Findley/Raymond Edwards. The game, "Galaxy" pits the operator of a spaceship against alien craft, as well as such variables as speed, time, and ammunition. No two games are the same! This game is described in *Galaxy Game for the 6800,* published by Scelbi Computer Consulting, Inc. $14.95

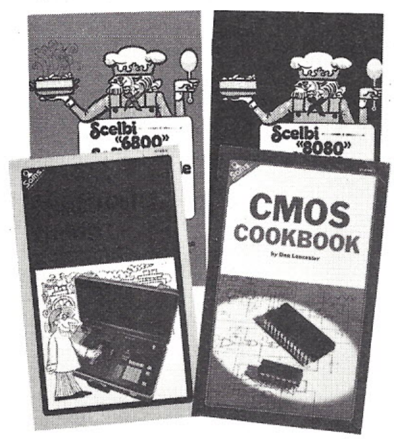
●**6800 SOFTWARE GOURMET GUIDE & COOKBOOK** If you have been spending too much time developing routines for your 6800 microprocessor, try the new book by Scelbi Computing and Robert Findley. This manual, *6800 Software Gourmet Guide and Cookbook* described sorting, searching, and many other necessary routines for the 6800 user. $9.95.

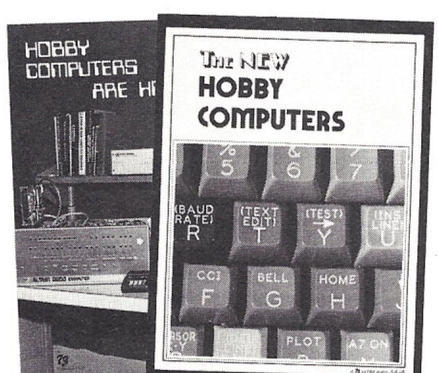
●**8080 SOFTWARE GOURMET GUIDE AND COOKBOOK** If you have been spending too much time developing simple routines for your 8080, try this new book by Scelbi Computing and Robert Findley. This manual, *8080 Software Gourmet Guide and Cookbook* described sorting, searching, and many other routines for the 8080 user. $9.95

●**CMOS COOKBOOK** by Don Lancaster, pub. Howard W. Sams Company. Another winner from Don Lancaster, author of the famous *RTL* and *TTL Cookbooks.* The *CMOS Cookbook* details the application of CMOS, the low power logic family suitable for most applications presently dominated by TTL. The book follows the style of the original Cookbooks. Eight chapters cover all facets of CMOS logic, and the work is prefaced by 100 pages detailing the characteristics of most CMOS circuits. The *CMOS Cookbook* is required reading for every serious digital experimenter. $9.95

●**HOBBY COMPUTERS ARE HERE** If you (or a friend) want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fundamentals and explains the circuits, the basics of programming, along with a couple TVT construction projects, ASCII-Baudot, etc. This book has the highest recommendations as a teaching aid for newcomers. $4.95.







●**SCELBI'S FIRST BOOK OF COMPUTER GAMES** Need a game for your 8008 or 8080 microprocessor? Try *Scelbi's First Book of Computer Games for the 8008/8080* which described three popular games, "Space Capture", "Hexpawn", and "Hangman". Complete flowcharts, logic description, program listing, and instructions are provided. A must for the game freak! $14.95.

●**THE STORY OF COMPUTERS** by Donald D. Spencer is to computer books what *Dick and Jane* to novels . . . extremely elementary, gives the non-computerist a fair idea of what the hobbyist is talking about when he speaks computer lingo. Attempts to explain what computers are and can do to a spouse, child or any un-electronics-minded friend. $4.95.

●**MICROCOMPUTER PRIMER** by Mitchell Waite and Michael Pardee, pub. by Howard W. Sams Company. If you are afraid to get involved with microcomputers for fear of not understanding them, worry no longer! The *MICROCOMPUTER PRIMER* describes basic computer theory, explains numbering systems, and introduces the reader to the world of programming. This book does not elaborate on specific systems or chips, but describes the world of microcomputing in "real world" terminology. There is probably no better way of getting involved with the exciting new hobby of microcomputing. $7.95

● **INTRODUCTION TO MICROPROCESSORS** by Charles Rockwell of MICROLOG Here is an ideal reference for the individual desiring to understand the hardware aspects of microprocessor systems. This book describes the hardware details of computer devices in terms the beginner can understand, instead of treating the micro chip as a "black box." Addressing schemes, registers, control, and memory are all explained, and general information about hardware systems is provided. Specific systems are not described and programming is only briefly discussed. *Introduction To Microprocessors* is a hardware introduction . . . and a good one. $17.50 US and Canada, $20 elsewhere.

●**THE NEW HOBBY COMPUTERS!** This book takes it from where "Hobby Computers Are Here" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards, a Baudot monitor/editor system, an audible logic probe for finding those tough problems, a ham's computer, a computer QSO machine . . . and much, much more! Everything of interest is there in one volume, ready to be enjoyed by you. $4.95.

Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud Book Dept., Peterborough NH 03458. Be sure to include check or detailed credit card information.
Note: Prices subject to change on books not published by 73 Magazine.

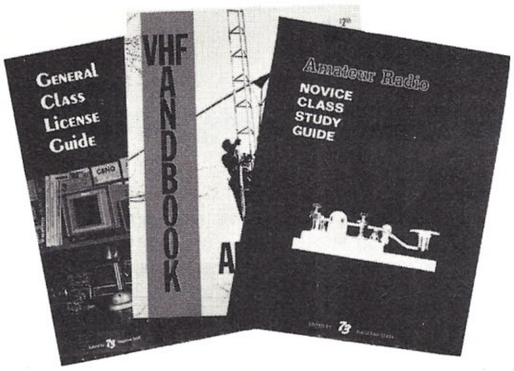
KB BOOK NOOK

Test Equipment Library

●**VOL. I COMPONENT TESTERS** Build your own test equipment and save a bundle (and have a lot of fun). Volume I of the 73 Test Equipment Library shows you how to build and use transistor testers (8 of 'em), three diodes testers, 3 IC testers, 9 voltmeters and VTVMs, 8 ohmmeter, 3 inductance meters, and a raft of other gadgets for checking temperature, crystals, Q, etc. $4.95

●**VOL II AUDIO FREQUENCY TESTERS** If you're into audio . . . such as digital cassette recording, RTTY, Baudot vs ASCII, SSTV, SSB, Touchtone or even hi-fi . . . you'll want to have this book full of home built test equipment projects. Volume II $4.95

●**VOL. III RADIO FREQUENCY TESTERS** This is of more interest to hams and CBers . . . test equipment you can build for checking out transmitters and receivers: signal generators, noise generators, crystal calibrators, GDOs, dummy loads . . . things like that. This is Volume III of the 73 Test Equipment Library $4.95



●**NOVICE STUDY GUIDE** This is the most complete Novice study guide available. It is brand new. This is not only invaluable for anyone wanting to get started in amateur radio, but also it is about the only really simple book on the fundamentals of electricity and electronics. And without your fundamentals down pat, how can you go on to really understand and work with computers? First things first. $4.95

●**GENERAL CLASS STUDY GUIDE** This book takes over on theory where the Novice book leaves off. You'll need to know the electronic theory in this to work with computers and you'll not find an easier place to get the information. It will also make getting your Tech or General license a breeze . . . then you can get on the ham repeaters and interconnect your micro with others. $5.95

●**VHF ANTENNA HANDBOOK** The NEW VHF Antenna Handbook details the theory, design and construction of hundreds of different VHF and UHF antennas . . . a practical book written for the average amateur who takes joy in building, not full of complex formulas for the design engineer. Packed with fabulous antenna projects you can build. $4.95

●**WEATHER SATELLITE HANDBOOK** Simple equipment and methods for getting good pictures from the weather satellite. Antennas, receivers, monitors, facsimile you can build, tracking, automatic control (you don't even have to be home). Dr. Taggart WB8DQT $4.95

●**SSTV HANDBOOK** This excellent book tells all about it, from its history and basics to the present state of the art techniques. Contains chapters on circuits, monitors, cameras, color SSTV, test equipment and much more. Hardbound $7, Softbound $5

●**RF AND DIGITAL TEST EQUIPMENT YOU CAN BUILD** RF burst, function, square wave generators, variable length pulse generators — 100 kHz marker, i-f and rf sweep generators, audio osc, af/rf signal injector, 146 MHz synthesizer, digital readouts for counters, several counters, prescaler, microwavemeter, etc. 252 pages. $5.95

●**WHAT TO DO AFTER YOU HIT RETURN** PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, muchly illustrated. Lunar landing, Hammurabi, King, Civil 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $8.00.

●**101 GAMES IN BASIC** Okay so once you get your computer up and running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you, from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.

●**BASIC** by Bob Albrecht, etc. Self-teaching guide to the computer language you will need to know for use with your microcomputer. 324 pages. This is one of the easiest ways to learn computer programming. $4.95

●**TVT COOKBOOK** by Donald Lancaster, describes the use of a standard television receiver as a microprocessor CRT terminal. Explains and describes character generation, cursor control and interface information in typical, easy-to-understand Lancaster style. This book is a required text for both the microcomputer enthusiast and the amateur RTTY operator who desires a quiet alternative to noisy teletype machines. $9.95

●**TTL COOKBOOK** by Donald Lancaster. Explains what TTL is, how it works, and how to use it. Discusses practical applications, such as a digital counter and display system, events counter, electronic stopwatch, digital voltmeter, and a digital tachometer. 336 pages; 5½ x 8½; softbound. $8.95

Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud Book Dept., Peterborough NH 03458. Be sure to include check or detailed credit card information.

Note: Prices subject to change on books not published by 73 Magazine.

INTRODUCING THE COMPLETE CHALLENGER SYSTEM.

OSI's DREAM MACHINE

is a totally integrated computer system.

Imagine a system complete with terminal, CPU, memory, floppy disk, software, and all the little necessities to make it work together immediately. Now imagine this complete system available not only fully assembled, but priced much lower than anyone else's kit. What you are dreaming of is OSI's "new" Challenger System!

In the configuration shown above, the Challenger includes everything an end user needs for a complete small computer system. All you add is 110 VAC power and a desk to put it on.

This fully-assembled system includes:

HARDWARE:

OSI Challenger 65 with 16K RAM, serial interface, system monitor PROM, and floppy disk bootstrap PROM.

OSI Challenger single drive floppy disk formatted for 250K bytes storage per diskette surface.

Stand-alone terminal and Sanyo monitor for 16 lines of 64 characters at 2400 baud (other terminal options are available).

And all interconnecting cables!

SOFTWARE:

2 diskettes containing over 100,000 bytes of software including OSI's powerful Disk Operating System with variable length sectors. **6502 DISK BASED RESIDENT ASSEMBLER/EDITOR!** A totally interactive Assembler/Editor which assembles up to 600 lines a minute and is completely compatible with MOS Technology's Cross Assembler format. This program also contains a powerful disk-based line editor with commands for general text editing. **OSI'S EXTENDED MONITOR:** A powerful machine language debugging and utilities package including a Disassembler which is format compatible with the Assembler! **OSI 6502 8K BASIC FOR DISK BY MICROSOFT:** This powerful BASIC has all the features of Altair® 8K BASIC for the 8080 plus higher speed and disk storage. And it comes complete with a BASIC program library.

DOCUMENTATION AND SUPPORT:

We include over 600 pages of hardware, software, programming, and operation manuals. The Challenger is based on the well-proven OSI 400 system. The over 2,000 OSI 400s and Challengers now in use assure continuing hardware and software support for this system for years to come!

EXPANDABILITY:

The Challenger System can now be expanded to 192K of RAM and 16K of I/O and ROM. There are over 13 accessory boards including A/D, D/A, parallel and serial I/O, cassette interfaces, a dual drive floppy, a video graphics display, several RAM and PROM boards, and multiple-processor configurations.

APPLICATIONS:

The Challenger system is complete, fully assembled and configured so that the Disk Operating System can be booted in immediately on system power-up. Even a relatively inexperienced operator can have a complex BASIC program on-line just seconds after the system is turned on. The ease of use, high reliability, and large library of standard BASIC applications programs make the OSI Challenger System the first practical and affordable small computer system for small business, educational institutions, labs, and the personal computerist.

PRICES:

Challenger System, complete as stated above with terminal and monitor

$2599.00

As above without terminal. Specify RS-232 or 20ma loop and baud rate

$2099.00

IMPORTANT NOTE:

One of the most important features of the Challenger System is that it is not really "new". OSI has been delivering the basic circuitry of the Challenger since November 1975 and the floppy disk since June 1976. The only thing new is the total integration of the components as a complete, simple to use, fully-assembled, small computer system.

For more <u>free</u> information and the address of the OSI Computer Dealer or representative in your area, write to: OSI; Dept. S; Hiram, Ohio 44234 or enclose $1.00 for the full OSI catalog which contains kits from $134 and fully assembled computers from $439.

OSI

Ohio Scientific Instruments

11679 Hayden Street, Dept. S, Hiram, Ohio 44234

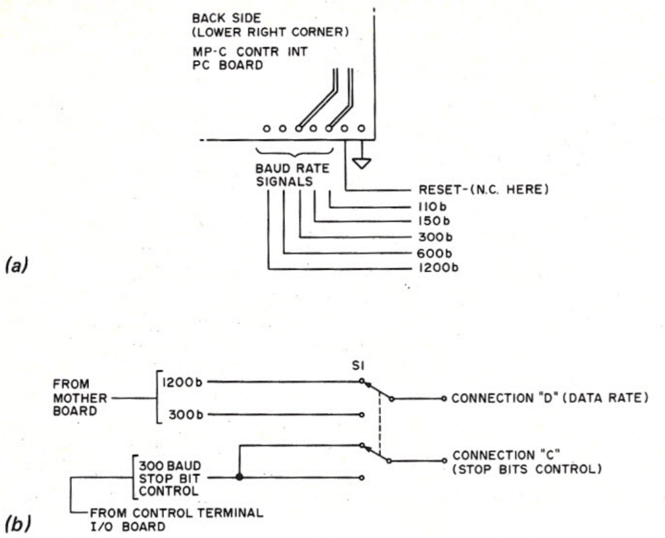
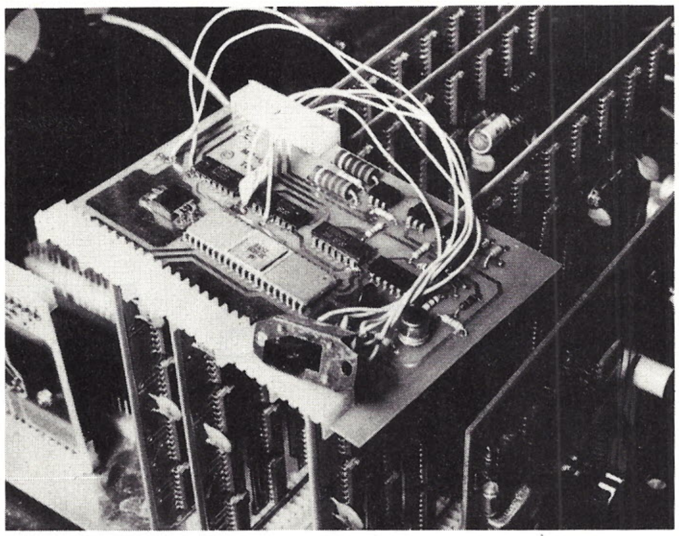


Fig. 1. (a) Showing pins on back of MP-C control interface board where baud rate timing signals are obtained. (b) Schematic of conversion. Note for 110 baud operation S1 must be DPDT and connected to give 2 stop bits (at connection "C") when 110 baud is selected.



SWTP 1200 baud converter in place on 6800 micro. Note switch allowing the device to select baud rate.

Jim Huffman
Hufco
PO Box 357
Provo UT 84601

Speed Up Your 6800

This handy little conversion should pretty well explode the myth that the SWTPC 6800 computer system control interface will only operate at 110 or 300 baud. This means that if your cassette interface has the ability to run at higher speeds it will now be able to do so quite easily while using routines nestled in MIKBUG ROM to oversee the loading and unloading process. There's probably no easier way for you to operate 1200 baud on your Southwest Tech 6800 system.

The advantage of running the control terminal at 1200 baud is obvious: You don't need another terminal to do the high speed communicating while the control terminal sits over in the corner waiting to tell the program to go or stop.

I had been using my SWTPC 6800 computer system for months, agonizing as the cursor slowly jiggled across the screen at 110 baud. At last, after three months of waiting, my baud rate option board arrived. I was now able to run my CT1024 terminal at speeds of 110, 150, 300, 600, and 1200 baud. Running the CT1024 at 1200 baud was no problem, merely write an output routine for the 6800 that supports a 1200 baud terminal. A little software UART, using one bit of a parallel output port as serial out and using another bit as a serial Teletype in.

All was well until the day my 8K BASIC arrived on cassette tape with no program listing. It was obvious I was going to have a hard time going into the 8K BASIC program and changing the MIKBUG addresses around to suit my own 1200 baud output system, so I decided to convert my Southwest Tech products control terminal output to 1200 baud (since it had a "hardware" bit-rate timer built right in) and boy, was it easy!

Conversion should take you less than fifteen minutes and the only part required is optional: a slide switch. As shown in Fig. 1, conversion is very simple. It merely consists of taking the 1200 baud already floating around on the Southwest Tech mother board and running it up to the normal 110 or 300 baud input to the clock generator on the Control Terminal I/O board. Note too, you can choose any of the other baud rates available on the mother board. My slide switch was installed as shown in the schematic to operate my control terminal at either 300 baud or 1200 baud so that I could load the Kansas City Standard 300 baud cassettes into my processor. The tape recorder used in my system is an Educassette model CC-1 unit that's capable of up to 4800 baud. Operating this system at 1200 baud, I find that I do not even need a cassette clock input to the Southwest Tech computer (precludes using any other recorder, of course) because I have yet to drop out a single bit in all the programs that I have loaded using this system. The cassette system is operating only at one fourth of its potential speed; therefore, slewing errors and dropout errors are virtually eliminated.

To this point, the only problem that I have encountered has been the ability of the cursor control option board on the CT1024 terminal to keep up with itself at 1200 baud; however, I suspect I would have this trouble any time I tried driving the terminal from a 1200 baud source, whether or not that source was the slightly modified SWTPC 6800 system. ■

Greg Pickles
PO Box 2276
Goleta CA 93018

Who's Afraid of RS-232?

...data communications explained!

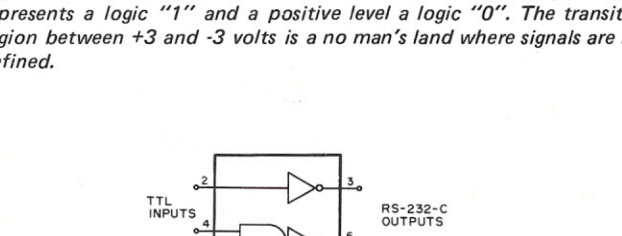


Fig. 1. RS-232 voltage levels. Levels are inverted so that a negative level represents a logic "1" and a positive level a logic "0". The transition region between +3 and -3 volts is a no man's land where signals are not defined.

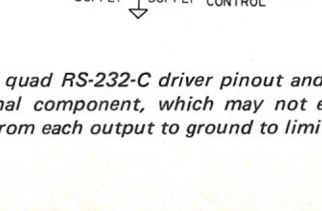


Fig. 2. MC1488 quad RS-232-C driver pinout and functional diagram. The only external component, which may not even be needed, is a small capacitor from each output to ground to limit the slew rate.

Who's afraid of RS-232 interfacing? Well, I can't think of a reason why anyone should be after reading Greg's discussion of the subject. You know, there are a lot of other interfacing considerations which need definition and explaining such as Greg has done here. Usually the designers who work with these things on a day-to-day basis take them for granted so much that they never think to sit down and write about them. I'm hoping we'll see more . . . and I hope they're as well done as this one. — John.

Sooner or later every computer hobbyist, whether he likes it or not, will run up against the omnious RS-232 interface. When this inevitable collision takes place, the hobbyist has little to go on but the odds and ends he can pick up hobnobbing around the local computer store. If you have come upon this savage foe, read on and discover what his true nature is and a few tips on how to tame him. If you have not yet encountered RS-232, take a friendly suggestion and read on — it never hurts to be prepared.

RS-232 Defined

RS-232 came into existence when the Electronic Industries Association (EIA for short) defined standard voltage and impedance levels for the transmission of digital data. This standard has allowed hundreds of different devices from different manufacturers to communicate with each other, usually by merely plugging them together. RS-232 is defined for serial communication between a modem and some other device, however the voltage level definitions are often used in other interfaces as well. My personal brush with RS-232 came when I was designing a dual serial interface which needed to talk to both a modem and a Teletype.

RS-232 comes in two varieties, B and C. The RS-232-B standard has been around longer and was designed for equipment which usually had considerably larger signal swings than we find today. The primary difference between RS-232-B and RS-232-C is the range of voltage levels allowed for a logic "1" and a logic "0". The voltage levels are shown in Fig. 1. For RS-232-B the permissible range for a logic "1" is -5 to -25 volts and the range for a logic "0" is +5 to +25 volts. For RS-232-C the ranges are -5 to -15 for a "1" and +5 to +25 for a "0". At a RS-232 receiver, any signal in the appropriate range will be recognized as a valid logic level. It is not necessary for the "1" and "0" levels to be symmetric about zero as long as they fall within the specified ranges. Notice that the logic levels are inverted from what we usually expect (i.e. a "1" is represented by a negative level and a "0" by a positive level). This is a potential point of confusion but just remember that up is down and everything will work out fine.

Since you are most likely to encounter only RS-232-C, the electrical specifications are given for it in Table 1.

Most of the electrical specifications are self-explanatory, however a couple of notes may be in order. The limitation on slew rate (the maximum rate at which the RS-232 signal may change) is concerned with the problem of cross talk between conductors in a multi-conductor cable. The faster the transition, the more coupling will take place between two conductors in close proximity. For many applications this is not a problem for the hobbyist because cable runs are short (cross talk is also an increasing function of the length of the cable).

These electrical specifications are often held to rather loosely by equipment that claims to be RS-232 compatible. For instance I have used a Teletype that has been converted to RS-232 which was quite happy with an input signal range of +.5 ("1") to +5 ("0") volts. In some cases you may be able to get away with nonstandard levels, but it takes very little extra effort to meet the standard and assure compatibility with any RS-232 device.

Special Purpose ICs for Conversion

Now that we know what RS-232 is supposed to be, how do we convert between it and the internal logic levels of our computer? Since most machines hobbyists use have TTL levels floating around inside them, let's consider only circuits that interface between RS-232 and TTL. For other logic levels, the concepts are the same but the circuits may need to be modified a bit to make them work properly.

There are a host of integrated circuits that are designed for the job of interfacing to RS-232, so let's look at a few. Motorola makes a pair of ICs that are worthy of consideration. The MC1488 is a quad RS-232-C line driver. (Pinout and functional diagram are given in Fig. 2.) This IC is very easy to use and allows one level of gating to be done in the IC for three of the four drivers. Typical power supply voltages are +12 and -12 volts with +15 and -15 volts maximum. The MC1488 requires an external capacitance of 330 pF on the output of each driver to meet the slew rate specification of 30 volts/micro-sec. If the cabling and receiver do not have a capacitance greater than 330 pF, an external capacitor should be connected from each output to ground in order to meet the spec.

The MC1489 is a quad RS-232-C line receiver (pinout and functional diagram are given in Fig. 3) which allows external control of the threshold voltages for each receiver. Pins 2, 5, 9, 12 can affect the threshold voltages (input levels at which the output changes state) by connecting them to a power supply through a resistor. For example, with a +5 volt supply and a 5kOhm resistor, the output will be a "1" for any input voltage below about -2.7 volts and a "0" for any input voltage above about -1.5 volts. With the control pin not connected an input of less than about +.8 volts will give a "1" while an input of greater than about +1.9 volts will give a "0". The difference of about 1 volt between

driver output levels with	logic "0" +5 to %1
driver output levels with 3kOhm to 7kOhm load	logic "0" +5 to +15 volts logic "1" -5 to -15 volts
driver output voltage with no load	-25 to +25 volts
driver output impedance with power off	greater than 300 Ohms
output short circuit current	less than .5 Amp
driver slew rate	less than 30 volts/micro-sec
receiver input impedance	between 3kOhms and 7kOhms
allowable receiver input voltage range	-25 to +25 volts
receiver output with open circuit input	logic "1"
receiver output with 300 Ohms to ground on input	logic "1"
receiver output with +3 volt input	logic "0"
receiver output with -3 volt input	logic "1"

Table 1. RS-232-C electrical specifications.

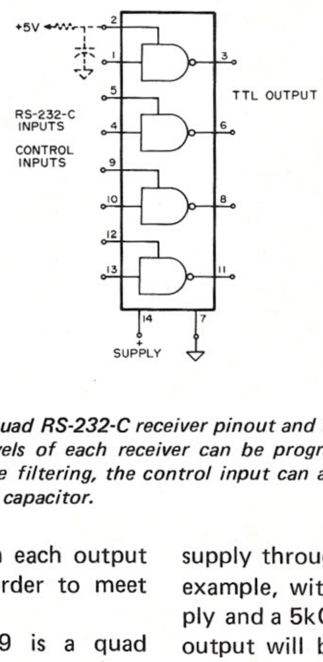


Fig. 3. MC1489 quad RS-232-C receiver pinout and functional diagram. The threshold levels of each receiver can be programmed by a single resistor. For noise filtering, the control input can also be bypassed to ground by a small capacitor.

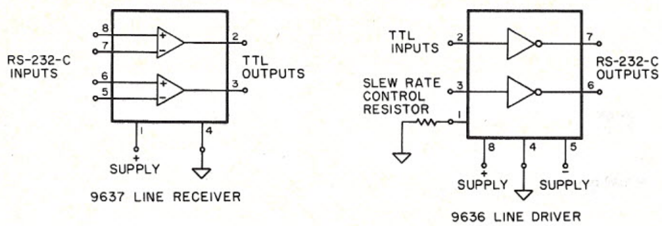


Fig. 4. 9636/9637 RS-232-C driver/receiver ICs pinout and functional diagrams. The slew rate of both drivers in the 9636 can be controlled by a single external resistor to ground. 8-pin DIPs save board space where only two lines are needed.

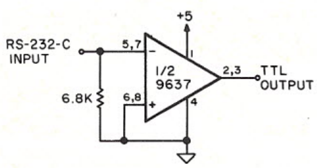


Fig. 5. RS-232-C receiver based on the 9637. The input resistor is necessary to assure an input impedance within the RS-232-C specifications.

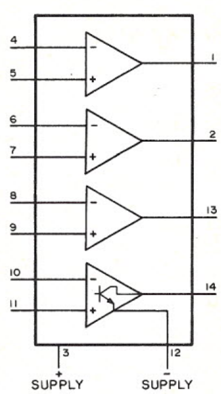


Fig. 6. Pinout and functional diagram of the LM339 quad comparator. Inputs are high impedance and outputs are an open collector transistor switch.

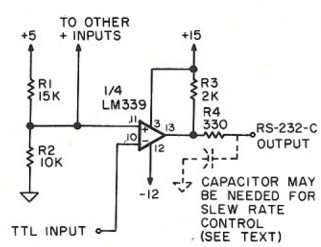


Fig. 7. RS-232-C driver based on the LM339. Usable to over 50 Kbaud. Unsymmetrical supply voltages give an approximately symmetrical output swing when loaded by 3kOhms.

the thresholds for "1" and "0" is called hysteresis (from the Greek word for shortcoming, hysteresis refers to a retardation of an effect as the cause of that effect is changed). The hysteresis gives a margin of noise immunity to the receiver because the input must change by a significant amount (over a volt) before the output will reflect the change. Additional noise immunity can be had with the MC1489 by putting a capacitor to ground on the control pin. As the capacitor is increased, the response to short pulses (the usual form of noise) is decreased. A capacitance of 500 pF will cause the receiver to ignore pulses of up to 6 volts whose duration is less than 800 nsec.

If four drivers or receivers are a few more than you need, you can use the Fairchild 9636/9637 ICs. The 9636 is a dual line driver and the 9637 is a dual differential line receiver. These ICs were designed to interface to RS-232-C with a minimum of external components and a further space savings is their 8-pin DIP packages. The pinout and functional diagrams of these ICs are given in Fig. 4.

The 9636 driver requires only power supplies (typically +12 and -12 volts) and a programming resistor from the waveshape control pin to ground. The programming resistor allows the rise and fall times of the output waveform to be varied from less than 1 micro-sec (10kOhms) to nearly 100 micro-sec (1 Megohm). The output voltage levels are fixed in the ranges -6 to -5 volts for a "1" input and +5 to +6 volts for a "0" input. A little elementary calculation reveals that a rise and fall time of .36 micro-sec will just meet the maximum slew rate specification of RS-232-C. Any programming resistor between 10kOhms and 1 Megohm will meet the specification with the larger values useful in limiting cross talk and noise generation problems.

The 9637 receiver is equally easy to use, requiring only a 5 volt power supply and an input termination resistor to provide the proper impedance level. As shown in Fig. 5, the positive input is connected to ground and the negative input to the source of RS-232 signals. The resistor from the negative input to ground assures that the input impedance will be less than the 7kOhms maximum specified for RS-232-C.

An Alternative to Special ICs

These commercial ICs are very useful in converting between RS-232-C and TTL with a minimum of components and space. Why, then, would anyone even think to suggest alternative techniques for accomplishing the conversions? Two main reasons quickly come to mind. First, the commercial ICs are somewhat expensive, costing over a dollar per receiver or driver function. Just as important as the cost factor is the relative scarcity of these ICs in the marketplace usually frequented by the hobbyist (surplus and mail-order dealers). My solution to both these problems was to design circuits which perform the desired conversions using easily obtained and inexpensive parts. The cost per function for these circuits is less than 50¢.

The RS-232 driver is based on a quad comparator IC, the LM339. The LM339's output stage is an open collector NPN transistor which functions as a switch. When the difference of the input voltages $V^+ - V^-$ is positive, the switch is open (the transistor turned off). If the difference becomes negative, the switch closes connecting the output to the negative supply voltage. Fig. 6 gives the pinout for the LM339.

The schematic diagram of the driver circuit is given in Fig. 7. The power supply voltages for the driver were chosen so that with the minimum load impedance of 3kOhms we get approximately symmetrical output levels. R1 and R2 form a voltage divider which provides a reference level of about 2 volts to the positive input of the comparator. This reference can be used for as many driver circuits as you want so the divider need not be repeated. The TTL input is applied to the negative input of the comparator. When the TTL input is at "0", it will be less than 2 volts and the output of the comparator will be connected to the +15 volt supply through R3. This supplies a high voltage out for a "0" input. If the TTL input is at "1", it will be higher than 2 volts and the output of the comparator will be connected to the -12 volt supply. R4 in series with the output satisfies the requirement for greater than 300 Ohms output impedance.

Fig. 8 shows the RS-232 output with a 3kOhm load. The output slew rate is just a bit too fast to meet spec (about 35 volts/micro-sec) but usually the capacitance of the cable and receiver will slow things down enough to be within spec. If you have some concern about the slew rate, putting a small capacitor (about 200 pF) from the output to ground will assure meeting the spec. Fig. 9 shows the output with a 270 pF capacitor in parallel with the 3kOhm load. This gives a slew rate of about 20 volts/micro-sec. At a data rate of 10 kilobaud, the effect of the capacitive load on the waveform is very small. There should be no problem when driving a highly capacitive line: With a capacitive load of 1000 pF (the equivalent of about 100 feet of cable) the output waveform was still very good.

So now that we have RS-232 output, how do we listen to what comes in? A one-transistor receiver circuit is shown in Fig. 10. If the RS-232 input is in its positive voltage state, the transistor will be biased on through R1 and the TTL output will be pulled down to ground ("0") level. If the input is in its negative state, the base of the transistor will be held at -.7 volt by the diode clamp, D1. With a negative voltage on the base of the transistor, it is turned off and the output is connected to +5 volts by R2. The clamp is necessary to protect the base-emitter junction of the transistor from the reverse breakdown which would occur if the full negative voltage were applied to it. R2 can easily supply the input bias current to three standard TTL inputs without dropping below the minimum voltage for a "1". If a larger fanout is needed, R2's value can be lowered somewhat at the expense of a higher current drain when the output is in the low state. The receiver input impedance (6.8kOhms) is in the higher end of the acceptable range so that it puts a minimum load on the data line. With either an open circuit or grounded input, the transistor will not be turned on and the output will be in the "1" state.

If we connect the output of the RS-232 driver to the receiver we observe the waveform shown in Fig. 11. At the 10 kilobaud data rate shown the receiver works fine, but at higher rates some trouble may occur because of the relatively slow rise time of the output. This rise time can be improved by reducing R2.

The 25 Pin D Connector

Many devices that use RS-232 interfaces also use some form of the RS-232 standard for the 25-pin D connector. The standard was written for the interconnection between a modem and some other piece of equipment such as a computer or video terminal. In the following description I will try to point out which lines are most often used for the interconnection. It is important to keep in mind that devices which claim to be RS-232 compatible often use only some of the defined signals. For this reason it is very helpful to have some informa-

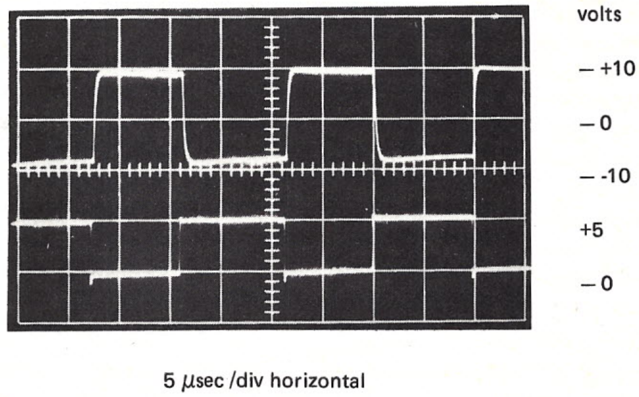


Fig. 8. Output (upper trace) of the circuit in Fig. 7 with TTL input (lower trace) and a 3kOhm load. Output slew rate is about 35 volts/micro-sec. Data rate is about 50 kbaud.

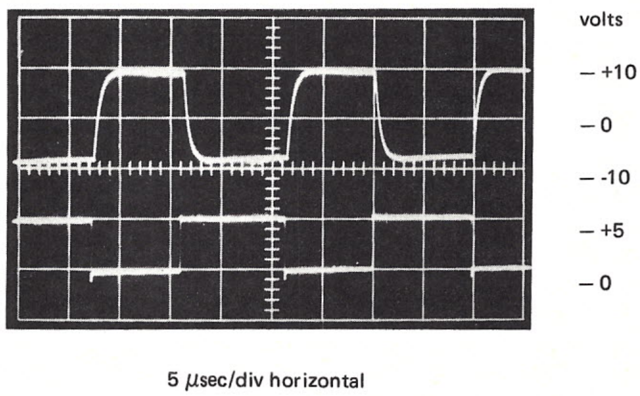


Fig. 9. Output (upper trace) of the driver in Fig. 7 with a 3kOhm load shunted by 270 pF. Slew rate is about 20 volts/micro-sec. Data rate is about 50 Kbaud. Lower trace shows TTL input.

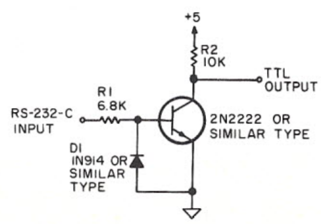


Fig. 10. Single transistor RS-232-C receiver. The transistor and diode are not critical; most other silicon types can be substituted. For a faster rise time or more drive capability, lower the value of R2.

tion on the connection scheme of the device before trying to hook it up. You should also be aware that some equipment uses the pins which are not necessary for its operation in a nonstandard manner. A good example is the common practice of using only pins 1-8 as they are defined by RS-232 and some of pins 9-25 for a current loop transmitter/receiver. Table 2 gives a summary of the pin definitions for the RS-232 connector.

Pin 1 is a protective ground line. It connects together the chassis of all pieces of equipment. Often equipment with motors and other rotating machinery can build up a sizable static charge while in operation. This protective ground line allows the charge to be bled away without affecting any of the signal lines. Pin 7 is the signal ground line which provides the ground level for the logic circuitry associated with the interfaces. Both pins 1 and 7 are commonly used.

Pin 2 is the data transmitted from the data terminal to the modem. The data is transmitted serially with the standard RS-232 levels. Pin 3 is used for sending the information received by the modem to the data terminal for display or other use. These two pins are obviously necessary in any bidirectional interface.

Several pins are defined to indicate the state of the modem. All of the signals used for handshaking and control are negative logic. A "0" (positive voltage) occurs when the associated condition is true and a "1" (negative voltage) when the condition is not true. Pin 6 is the data set ready indicator. This line is used for hardwired modems that can operate in either the voice or data mode. For the data mode, pin 6 is at "0" and for voice transmission, pin 6 is at "1". If the modem can be used for data only, the line is always held at "0". Pin 8 indicates that the modem has detected a carrier tone from the modem at the other end of the line and that the link is now usable. Pin 22 indicates that the modem has detected a ringer signal on the telephone line. This is used by equipment that must be able to automatically answer incoming calls.

Pin 20 is an indication from the data terminal that it is in the "on line" mode. If the line is at "0", the terminal is "on line" and data from it is to be transmitted by the modem. A "1" indicates that the terminal is "off line" (or in the local mode) and no data is to be sent.

Pins 4 and 5 are used for a handshaking between the terminal and modem. Pin 4 (request to send) goes to "0" when the terminal wishes to send data. The modem then returns a "0" on pin 5 (clear to send) when it is ready for the terminal to transmit. Often this full handshaking is not used and static "0" levels are put on the lines.

Pins 15, 17, and 24 are used by modems that transmit at relatively high rates (1200 and 2400 baud, for example) for data synchronization. Pin 15 is a transmit bit clock. Some modems generate this clock internally and some require an external clock which is supplied to pin 24. The serial bit stream to be transmitted is applied to pin 2 with each bit occuring at the positive going transition of the clock. The first bit must appear at the first positive going transistion after a "0" occurs on the clear to send line (pin 5). Pin 17 is a receive bit clock which indicates each bit in the received bit stream by a positive going transition. Most of the equipment we, as hobbyists, deal with do not use these lines.

A brief recap shows that the signal lines most hobbyists will be concerned with are pin 1 (protective ground), pin 2 (transmitted data), pin 3 (received data), pin 4 (request to send), pin 5 (clear to send), and pin 7 (signal ground).

Conclusion

With these circuits it should be no problem to interface to a RS-232 compatible device. The special purpose ICs are very useful for saving space because they require a minimum of external components. However, for a little more space and about half the money, the other circuits will do the job and your peripherals will never know the difference. We have looked at the standard pin assignment for the 25-pin D connector and found that it really isn't so hard to get along with after all.

Hopefully by now RS-232 has ceased to be a nebulous entity and, while you may not be ready to befriend it, at least you can deal with it effectively. ■

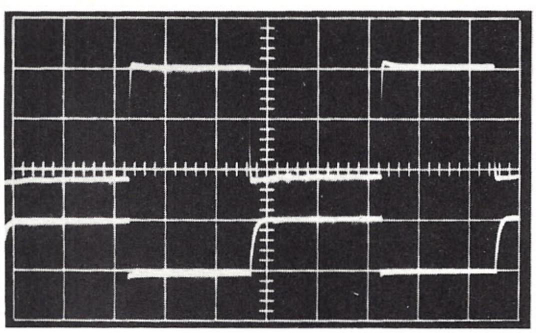


Fig. 11. Output (lower trace) of the RS-232-C receiver in Fig. 10 in response to the signal from the driver in Fig. 7. Data rate is about 10 Kbaud.

25 pin connector pin number	EIA RS-232 name	description
1	AA	protective ground
2	BA	data transmitted from terminal
3	BB	data received from modem
4	CA	request to send
5	CB	clear to send
6	CC	data set ready
7	AB	signal ground
8	CF	carrier detector
9-14	—	undefined
15	DB	transmitted bit clock, internal
16	—	undefined
17	DD	received bit clock
18-19	—	undefined
20	CD	data terminal ready
21	—	undefined
22	CE	ring indicator
23	—	undefined
24	DA	transmitted bit clock, external
25	—	undefined

Table 2. RS-232 connector pin definitions.

GET THE BEST FROM... SEALS

ORDER FORM [Write in each box the quantity of each part required]

KIT		ASSEMBLED	KIT		ASSEMBLED	KIT		ASSEMBLED
☐	4KROM	☐	☐	16KSC	☐	☐	68EXT-L	☐
☐	8KSC-Z	☐	☐	WWC	☐	☐	68WWC	☐
☐	68KSC	☐	☐	88 EXT	☐	☐	BBUC	☐
☐	8KSC	☐	☐	68EXT-S	☐			

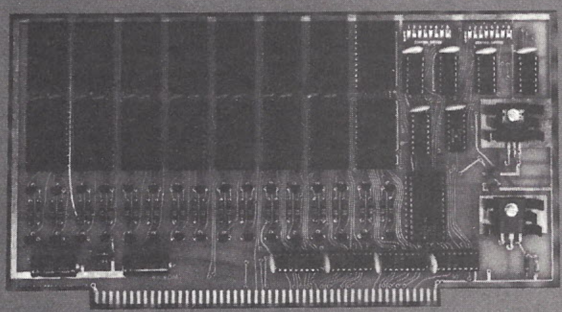
☐ ASSEMBLY AND OPERATING MANUAL $4.00

NAME ______________________
PLEASE PRINT OR TYPE
ADDRESS ______________________
CITY ____________ STATE ________ ZIP ________

SEND CHECK • MONEY ORDER • COD'S ACCEPTED • CREDIT CARDS

SEALS ELECTRONICS, INC
P.O. BOX 11651
KNOXVILLE, TN 37919
MOST ORDERS SHIPPED WITHIN 10 WORKING DAYS

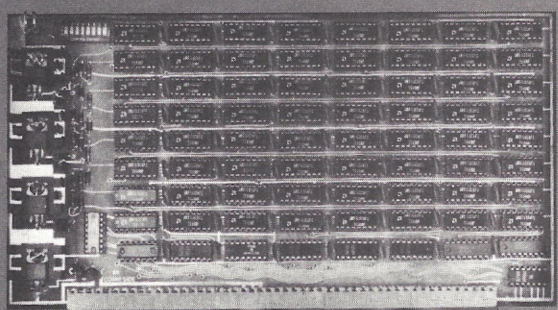
BANKAMERICARD — COPY ACCOUNT NUMBER FROM YOUR BANK AMERICARD — MY CARD EXPIRES
master charge — COPY ACCOUNT NUMBER FROM YOUR MASTER CHARGE — Copy number above your name on Mastercharge — MY CARD EXPIRES



4KROM [READ ONLY MEMORY]

- Will accept up to 16 ea. 1702-A or 5203 EProm Providing up to 4096 words of non-volatile memory for Boot Loads to Complete Programs.
- Programming Available at Factory for $3.00 per EProm when accompanied by binary formated tape.
- Each 1702-A has its own Vgg clocked for Low Power Consumption.
- Will work with the weakest power supply based S-100 buss computer.
- Switched Selected Address in 4K Blocks.
- Switch selected wait states so that even the slowest 1702-A can work in your system. 0-8 wait states.
- Solder Masked on both sides of PC Board.
- Component Screened on Component Side of PC Board.

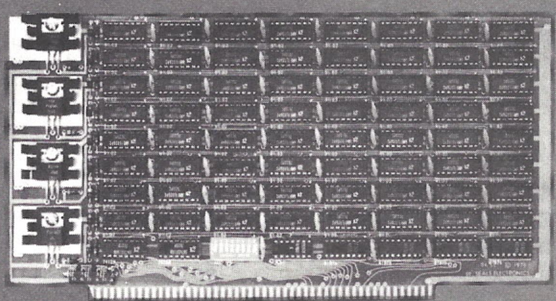
Kit Price: . . $119.00 Assembled Price: . . .$179.00



68KSC [STATIC MEMORY CARD]

- 8192 Words of Static Memory
- Access Time: 500 nsec. (250 nsec on request)
- Memory Chip 91L02 APC
- Battery Standby
- Address Selected 8 Ea. SPST Dip Switch
- Low Power
- All Lines Buffered.
- All IC's with sockets
- Solder Masked on both sides of PC Board.

Kit Price: . . $269.00 Assembled Price: . . .$359.00



8KSC-Z [STATIC MEMORY CARD]

- ALTAIR®, IMSAI®, and S-100 buss compatible.
- Access Time: 250 nsec max.
- Zilog Speed Compatible up to 4 mhz.
- Memory Chip: 2102LHPC or 2102-2
- Battery Standby: >1.5 to 4 volts<
- Address Select: 8 ea. Spst. Dip Switch.
- Wait States: None
- Current Reg.: Less than 200 ma per 1K
- All Address, Control, and Data out lines fully buffered.
- All IC's supplied with IC Sockets
- Solder Masked on Both Front and Back of P.C. Board.

Kit Price: . . $295.00 Assembled Price: . . .$395.00

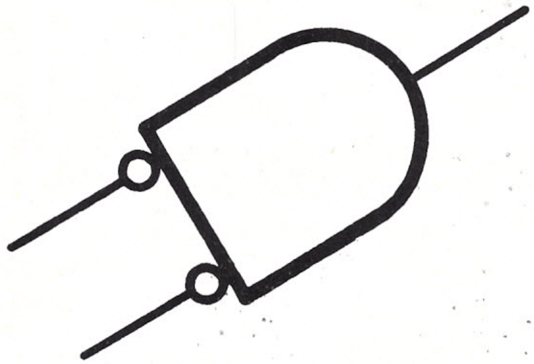
OTHER SEALS ELECTRONIC PARTS AVAILABLE

Item No.	Description	Kit Price	Assembled Price
8KSC	8K Static Memory Card 500 nsec	$269.00	$369.00
WWC	Wire Wrap Card	$ 37.50	$ 47.50
88 EXT	88 Extender Card	$ 29.00	$ 38.00
68 Ext-S	Extender Card (Small) SWTPC 6800 Compatible Products	$ 19.00	$ 25.00
68 Ext-L	Extender Card (Large) SWTPC 6800 Compatible Products	$ 29.00	$ 39.00
68 WWC	Wire Wrap Card	$ 35.00	$ 45.00
BBUC	Battery Back-Up Card	$ 55.00	$ 68.00

ORDER DELIVERY: FROM STOCK TO 10 DAYS

SEALS® ELECTRONICS, INC.
TELEPHONE # 615/693-8655 S22

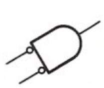
Peter A. Stark
P.O. Box 209
Mt. Kisco NY 10549



Is it High? - or Low?

. . . understanding logic design conventions

Undoubtedly there have been occasions when you hardware novices have been thrown into wonderment and confusion by the fact that logic gates can be used for either OR or AND functions (whew! . . . talk about confusion!). Or, perhaps you've questioned the use of those circles used on gates and flipflops? Have no fear . . . Pete Stark is here! – John.



Is it a NAND? Is it an OR? Is it a plane? Perhaps it's Superman? No! It's a NOR gate!

If you've seen logic symbols like this one and couldn't find them in the IC catalogs, then read on and find out why. I will take you through an introduction to digital logic circuits and then show you the special symbols used by the professionals.

When you first start experimenting with digital logic, you discover that logic circuits use ones and zeroes for signals. A signal is either present or absent, never both. We represent these by the symbols 1 and 0. In logic circuits, these signals are represented by voltages whose values depend on the type of circuit we use. For example, with TTL integrated circuits the voltages take on one of two values: a low value somewhere between 0 and perhaps 0.5 volts, and a higher value somewhere between 2.5 volts and 5 volts. With CMOS ICs the low voltage is usually very close to 0 volts, while the high voltage may be between 5 and 10 volts. The precise value of these voltages is not important, as long as they are somewhere in the range specified and not in between. Instead of referring to the precise voltage levels, we often just call them Low and High, or just L and H for short.

Now comes the confusing part. Logic signals are either 0 or 1, while the voltage levels are either L or H. Many people just assume that a logic 0 is always the Low voltage, while a logic 1 is always the High voltage. Most of the textbooks and introductory articles on digital circuits make that assumption and so most people think it's always so. But that's not true! Sometimes it's convenient to do just the opposite. In fact, professional designers often use both – they often change back and forth even within the same piece of equipment. One circuit may use a low voltage for a 0, while another circuit just a half inch away may use a high voltage for a 0. Why? Because it's convenient and perhaps even cheaper.

To understand what's going on, it helps to look at some of the buzzwords used by the experts. The symbols 0 and 1 are convenient when we are talking about a circuit used to represent binary numbers, such as the address lines of a computer which specify a binary address. On the other hand, many of the wires in a computer are used to describe the status of something, rather than a number. For example, there may be a line called BUSY, which is used to describe whether a printer is busy or not. Rather than talk about ones and zeroes, it's much easier to say that the line is either true or false. Generally TRUE is just another way of saying 1, and FALSE is the same as saying 0. The 1 and 0 are used to describe wires used to carry numbers, while TRUE and FALSE are used to describe wires carrying other information.

Now another word: We say a line is *active* if it is true. Going back to the printer we mentioned above, the printer BUSY line would be active when the printer is busy, and would be inactive, or not active, when the printer is not busy. Makes sense, doesn't it?

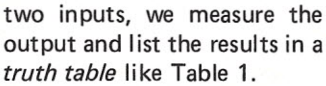
Now back to the confusing part. When it comes to designing that BUSY line and its circuitry, one designer might choose to assign the voltages in such a way that the line has a low voltage on it normally and the voltage goes high when the printer is busy. In other words, this line would be high when active, and low when not active. We would then call this line *active high.*

Another designer might do the exact opposite though. For various reasons, he might design the circuitry so that the line has a normally high voltage and the voltage goes *low* when the printer is busy. In this case, the line is low when active and high when not active. We would call this line *active low.*

Without going into specifics we have no way of choosing one way over the other. As long as everything is properly matched to this line, either way will work. Both ways are right, and both will work, though at times one may seem a little easier to understand than the other.

Let's recap. An *active high line* has a high voltage on it to represent a TRUE or 1 condition. An *active low line* does the exact opposite — it has a low voltage on it to represent a TRUE or 1 condition. Most textbooks and articles assume all lines are active high and assume that active low lines do not exist. On the other hand, if you look into a real piece of equipment you will find some of each.

Now let's apply this to real IC gates and other components. Suppose you get a 7432 TTL integrated circuit; in the catalog this is called a Quad 2-input OR Gate, meaning that it has four two-input gates in the same package. The symbol for each gate is this:



Knowing that logic circuits operate on low and high voltages, we put it on the lab bench and run some tests. Putting two voltages on the two inputs, we measure the output and list the results in a *truth table* like Table 1.

Input A	Input B	Output
Low	Low	Low
Low	High	High
High	Low	High
High	High	High

Table 1.

With two different input terminals, there are four possible combinations of inputs — both low voltage, both high, A low and B high, or B low and A high. From Table 1 we see that if either A is high OR B is high, then the output is also high. This gate is called an OR because its output goes high whenever either A *OR* B is high.

Now, a textbook or magazine article which only talks about active high lines would automatically replace every Low in the truth table by a 0, and every High by a 1. The resulting truth table would look like Table 2.

Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Table 2.

This is the way you always see OR gate truth tables in books. In words, the output is a 1 whenever either A is 1 *or* B is 1.

But suppose you give the High-Low truth table to a designer who happens to prefer active low circuits. He will do the exact opposite — he will put down 1 for every Low and 0 for every High. His truth table will look like Table 3 when he's done.

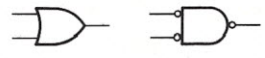
Input A	Input B	Output
1	1	1
1	0	0
0	1	0
0	0	0

Table 3.

If you're up on your logic circuits, you may recognize this as the truth table for an AND gate. It happens to be upside down from the way it's usually written so it may be hard to recognize, but in this case the output is a 1 only when both A is 1 *and* B is 1, so it behaves like a real AND gate.

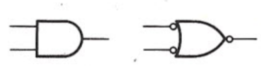
Now, how is it possible that the same gate — the same piece of hardware — can be both an OR and an AND? The answer is that what it does depends on how you use it. When used with active high inputs and outputs, it behaves like an OR. When used with active low inputs and outputs, it behaves like an AND.

Now back to the symbol, shown earlier. If the active-low designer uses this 7432 gate as an AND, wouldn't it be confusing if he drew it as an OR on his diagrams? Why not use a symbol that looks more like an AND? That's exactly what we do. Both of the following symbols may be used to describe the same 7432 gate:



The one on the left is the standard OR gate symbol, while the one on the right looks like an AND except that it has those little circles on the inputs and outputs. The circles stand for *active low.* They tell you that this gate behaves like an AND with active low inputs and active low outputs. One look at this symbol and you know what the gate does.

Let's do the same for a 7408 Quad 2-Input AND. Again, there are two possible symbols for the same gate:

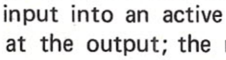


As before, the one on the left is the one we usually see — an AND gate with active high inputs and outputs (no circles). But the *same gate* behaves like an OR when the inputs and outputs are both active low, as shown by the second symbol. If you don't believe me, draw the truth table first with Highs and Lows and then substitute ones and zeroes for the two cases.

OK, let's look at inverters. An inverter changes a high voltage into a low voltage and vice versa. No doubt about that. But some books tell you that an inverter changes a 1 into a 0, and a 0 into a 1, and that may not be true. Look at Fig. 1, which describes what happens when a computer system is designed by a committee. On the left is a computer designed by a fellow who likes active high circuits; on the right is a printer designed by a chap who happens to prefer active low circuits. They both knew that they needed a BUSY line coming out of the printer to the computer, but they forgot to get together and agree on the voltages used.

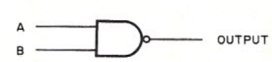
So the printer designer set things up so the line is *low* when the printer is busy; for him a low is a TRUE. On the other hand, the CPU (Central Processing Unit) designer needs a *high* when the printer is busy. (By the way, the printer designer called his output a $\overline{\text{BUSY}}$; the line above the word signifies that the line is active low.) What do you do? The answer is obvious — add an inverter between the two boxes, which will change the low into a high.

In this case, the inverter is changing a TRUE into a TRUE. We might also say that it is changing a 1 into another 1. The point is that an inverter doesn't always change a 1 into a 0. Sometimes an inverter is used to change an active low line into an active high line or vice versa. For this reason there are two symbols for an inverter also:



Both symbols have a circle but one has it at the output and the other at the input. They function exactly the same way but the left symbol would be correctly used to change an active high line at the input into an active low line at the output; the right symbol would be used to change an active low input into an active high output. (The right-hand symbol should be used in Fig. 1.)

Now we are finally ready to look at NAND and NOR gates. The 7400 NAND gate has the following symbol in the catalog:



The IC catalog also gives its truth table as in Table 4.

Input A	Input B	Output
Low	Low	High
Low	High	High
High	Low	High
High	High	Low

Table 4.

Looking at its symbol, you may start to suspect that this article has been leading up to a sneaky conclusion: Doesn't that look like an AND gate with active high inputs and active low output? Let's try that on the truth table – for the inputs replace each Low with a 0 and each High with a 1; for the output do the opposite, see Table 5.

Input A	Input B	Output
0	0	0
0	1	0
1	0	.0
1	1	1

Table 5.

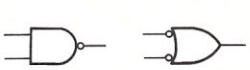
Eureka! We have a (weird) AND gate. Now you may see why people use a lot of active low lines – when you use 7400 NAND gates they turn up all the time.

Just so you don't miss it, let me warn you that you are about to be surprised. Watch what happens when you assume that the NAND inputs are active low, and the output is active high. Take the truth table, switch all the Lows and Highs, and out comes Table 6.

Input A	Input B	Output
1 (Low)	1 (Low)	1 (High)
1 (Low)	0 (High)	1 (High)
0 (High)	1 (Low)	1 (High)
0 (High)	0 (High)	0 (Low)

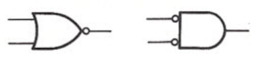
Table 6.

Right! The ones and zeroes are those of an OR gate. If we assume active low inputs and active high output, then the NAND gate is really doing an ORing. As it happens, there are two symbols for 7400 NAND gates:



The left one is the common one and shows that the gate does an ANDing with active high inputs and active low output. The one on the right shows that with active low inputs and active high output we have an OR function.

Now let's get back to the NOR gate. It too has two symbols:



When we think of the inputs as active high and the output active low, as in the left symbol, then we have an OR function. On the other hand, with active low inputs and active high output it does an ANDing. Just watch those little circles.

Circles are used for other ICs as well to denote active high or active low. For example, Fig. 2 (a) shows a 7473 JK flip-flop. The circle on the CP input tells you that the flip-flop triggers when the clock pulse (CP) goes *low*. The circles on the set (S) and reset (R) inputs tell you that you need a low on these inputs to set and reset. In Fig. 2 (b), the circles on the outputs indicate that the outputs go low when active. The 7441 is a Nixie tube driver, which grounds its outputs to turn on a given Nixie digit.

Figs. 3 through 5 show a few examples of how circles are correctly used; all of these are taken from an Intersil microcomputer using their IM6100 microprocessor. In Fig. 3 we have a circuit which is supposed to reset a 74C74 flip-flop when both DEV SEL *and* 64X3 go low. Both (a) and (b) in Fig. 3 are the same circuit, using a 4025 CMOS NOR gate, but (b) helps to explain that we are ANDing two active low signals. The output of the 4025 is an active high signal, which is then inverted into the active low needed to drive the reset input of the flip-flop.

In Fig. 4 we want the RESET signal to go low whenever either DEV SEL *and* 64X6 both go low, *or* when the switch is closed (so that point A goes high). From the description you see that we want to do an ANDing and an ORing, but a 4001 NOR gate is used. Fig. 3 (b) helps to explain that the first 4001 is ANDing two low signals, and the result is being ORed in the second 4001.

Fig. 5 gives another example. The IM6100 has two outputs called MEM SEL and CP SEL, which go low for memory select or control panel select, respectively. We want to generate a signal called SEL which goes low whenever either MEM SEL *or* CP SEL goes low. Despite the fact that we stressed the word *or* above, a 4011 CMOS NAND gate is used. In Fig. 5 (a), by skipping a few circles and using the wrong symbol for the 4011 NAND gate, we completely manage to hide its true function – ORing. But look at (b). By putting a few circles on the IM6100 outputs, we indicate right away that these outputs go low when active. Then the 4011 is drawn as an OR with active low inputs, which immediately lets you know that it is ORing the active low inputs. But since the 4011 has active high outputs, we need an inverter to make SEL active low as required. It may still be hard to understand, but when you become really familiar with the proper use of circles, it is a tremendous help.

There is only one fly in the ointment. Even among professional logic designers there are some who, while they are familiar with designing active low lines as well as active high, get careless with where they put their circles. They either put in extra ones all over the place or they skip a few. On top of that, when the art department redraws their diagrams nobody pays much attention to "those funny round things." And so you can't always trust the little critters to be in the right place. But when properly used they are great. ■

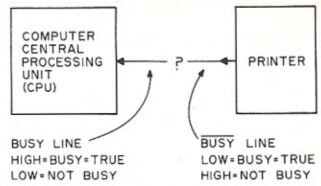


Fig. 1. Connecting an active low line to an active high line.

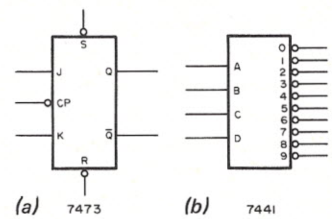


Fig. 2. Other common ICs.

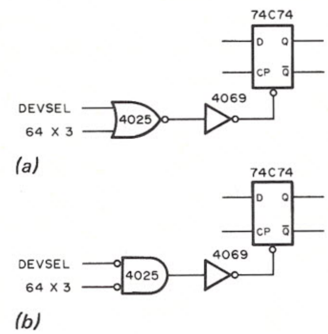


Fig. 3. CP Interrupt flip-flop from Intersil microcomputer.

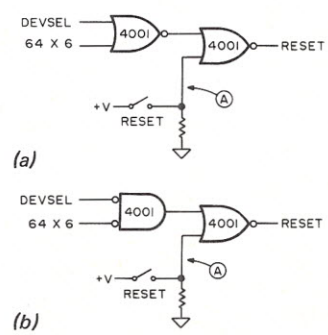


Fig. 4. RESET circuit from Intersil microcomputer.

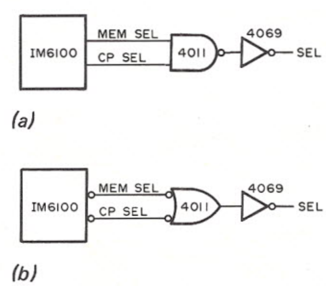


Fig. 5. Memory Select circuit from Intersil microcomputer.

Complete Control.

Introducing IMSAI 8048 Single Board Control Computer.

Complete Control System
Intel developed the world's first single chip microcomputer. IMSAI has built the system to put it to work. Instantly!

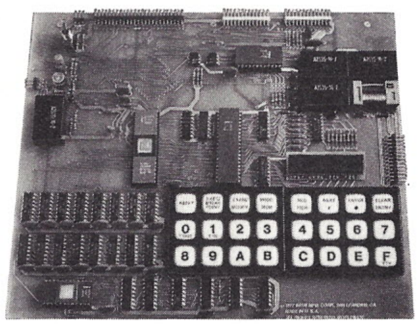
Presenting the IMSAI Single Board Control Computer. A complete programmable computer and hardware control system on an 8½" x 10" board.

Just plug in the power source (a 5V power supply or a 6V battery will do), and you can be controlling your home environment today!

The Chip Designed for Control
Intel designed the 8048/8748 single chip microcomputer with one thought in mind. Complete control. Everything you need is there: CPU, RAM, 1K ROM/EROM, I/O, timer/counter, interrupts . . . the works.

The Board That Implements It Instantly
There's a 24 pad hexadecimal keyboard and 9-digit LED hex display already on board. So you can start controlling the coffeepot and the sprinklers the minute you get the IMSAI home. Without adding anything but the connecting wires.

You can run almost any peripheral available. Not to mention the kids' electric train. The IMSAI Control Computer is



RS232 compatible. There are 12 quasi-bidirectional I/O lines with handshaking, and 14 more regular I/O lines, 5 heavy duty relays, and Teletype and audio cassette interfaces. All on one board.

There's already an extra 1K of RAM on board, plus sockets for another 1K of RAM and 2K of ROM/EROM. Still need more memory? The IMSAI 8048 allows expansion up to 64K of RAM off board.

Ultimately, the only limit to this system is your imagination.

Now, that's control.

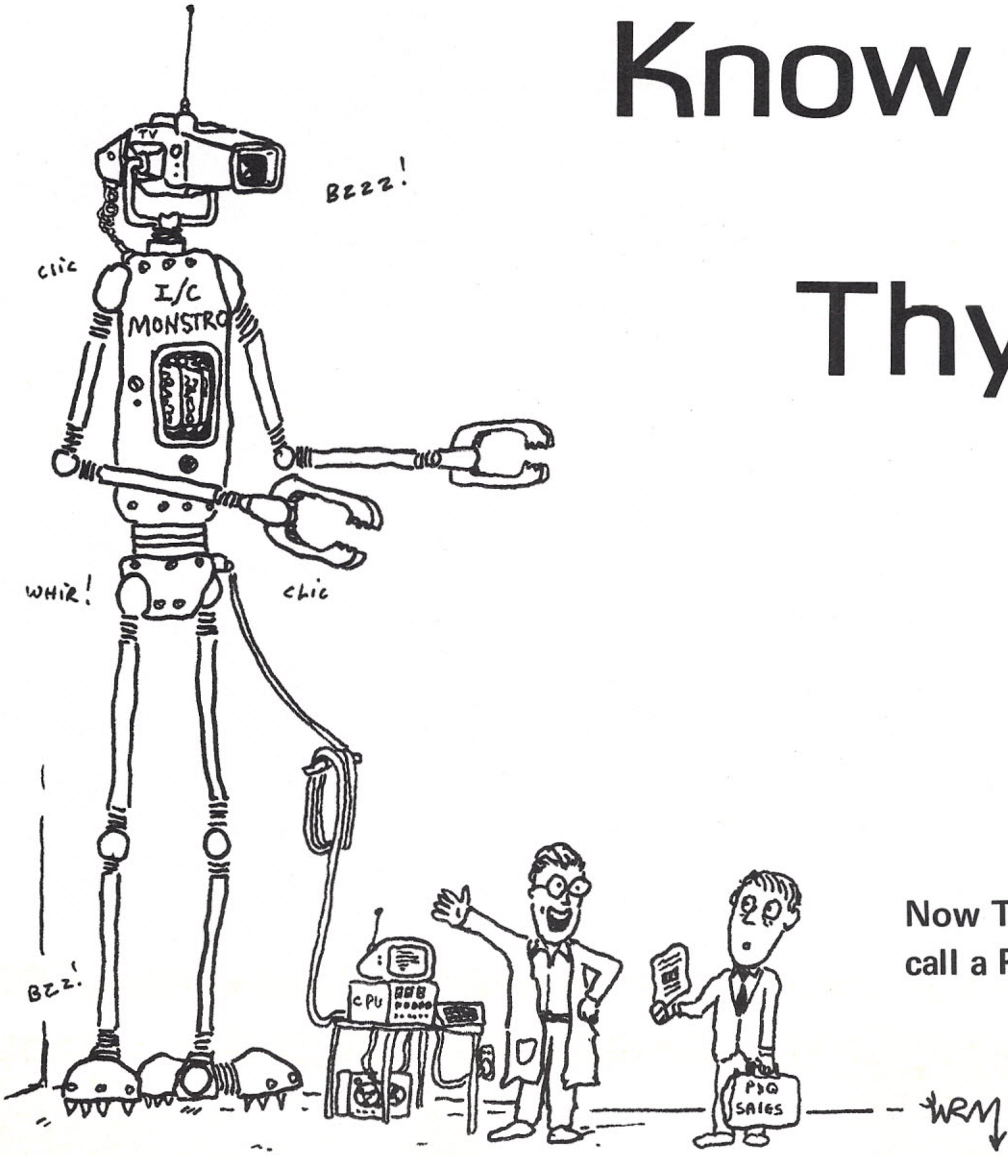
For instant control, use the coupon provided.

GENTLEMEN:
I want instant control!
☐ Send ROM Computer Kit $249*
☐ Send EROM Computer Kit $399
☐ Send assembled ROM Computer $299*
☐ Send assembled EROM Computer $499
☐ Send 5V power supply $99
*June delivery.
Check/MO enclosed. Amt. $ ______
Charge my: ☐ BAC ☐ M/C
______ Exp. Date ______
Sig. ______
☐ Send more information.
Name ______
Address ______
City ______
State/Zip ______

IMSAI®

IMSAI Manufacturing Corporation
14860 Wicks Blvd.
San Leandro, CA 94577
(415) 483-2093
TWX 910-366-7287

Know Thyself!



Now THAT'S what I call a Peripheral!

Ken Knecht
539 W. Addison
Chicago IL 60613

There are some good reasons why most of the manufacturers prefer to sell assembled units rather than kits. After reading of Ken's adventures with his machine you'll see why there are a lot of hobbyists who prefer this route also. But . . . there is an awful lot of satisfaction from getting that beast from a do-nothing condition to a working, blinking-lights computer-machine. – John.

It all started last fall when I decided I must have a computer. I told myself (I have no XYL to convince and besides, I'm easier to talk into doing things) to just stop and think how useful a computer would be. I could use it for all sorts of worthwhile things such as a burglar alarm for the apartment, keeping records, keeping an inventory of all sorts of things such as my record collection (which I haven't had time to listen to for years), my stamps (although I have a perfectly good card index of those), apartment contents for insurance (but what if they stole the computer?), electronic equipment design (but then what would I do with my HP-35?), and I was sure other sensible uses would occur to me later. But – deep down in my heart of hearts I knew what I was going to do with the computer – play games! I had heard all about Star Trek, card games such as poker and blackjack, lunar landing, and dozens more.

Taking the Plunge

So I ordered an Altair 8800 with 12K of memory, an ACR interface, BASIC language, an expander board, and De-bug (whatever that was I had a feeling I'd need it). Of course I bought everything in kit form – after all, I'd built my share of Heathkits, etc., hadn't I? And sooner or later I'd gotten them all to work – after a fashion.

For the uninitiated, as I was at the time, the computer of course is the power supply, CPU (central processing unit), front panel with rows of toggle switches and blinking

. . . confessions of a kit-builder

LEDs, and of course, the cabinet. To make it work you need memory; I bought 12K because the BASIC would use up half of it and the rest could be used for the programs. The 12K means it holds just over 12 thousand words (each word consists of 8 bits) or over 96000 bits. This seems like a lot of memory, but believe me, when you try to type in a BASIC program it goes fast. For example, each letter you type in takes up one word of memory, also each space, period, comma, etc. The ACR interface was to be used to connect a cassette recorder to the computer. I bought the BASIC on cassette so I needed this. The alternate possibility was to get the BASIC on a Teletype punched tape and use the same interface (I/O) board for the Teletype and for BASIC read in using the tape reader. I/O stands for input/output and you need one for every different type of equipment you want to interface with the computer. However, I didn't happen to have a Teletype sitting around and, what to do with all that used paper, not to mention buying new paper, didn't appeal to me. Also, Teletypes are rather slow and noisy animals. However, I have since had second thoughts about that. The expander board was to provide a place to plug in all the boards; the basic Altair only has four sockets and I had purchased five boards. We'll talk about the BASIC language a bit later.

Finally after about six weeks I got the computer and everything but the De-bug and the expander board, which were back ordered.

The Building (and Debugging) Begins . . .

It took me a week of evenings and about two cases of Pabst to put the computer, ACR board and one memory board together.

First of all, the book says, check the 100 contact sockets to be sure there are no shorts between the pins. OK, so far so good. How about that! Then fire it up and check the power supply voltages. Good grief, no 8 volts! Finally after a lot of checking and eventually tearing the power supply out of the computer I found a cold solder joint (the first of many!) on the diode bridge. Resoldered it, checked continuity and put it all back together. Fired it up again, still no 8 volts! Tear out the power supply again (I was getting good at it by now) and start checking further down the line. Finally found a jumper wire, one end not soldered, and by dint of great determination, avoiding contact with the PC board hole it was in. Soldered it and finally had all my voltages. Just a momentary lack of attention to detail I told myself, not believing a word of it, all the other thousands of solder joints will be OK.

After that I looked up the first provided machine language program (that's how sure of myself I was) to familiarize myself with it, so when I put in the memory board and CPU board and they worked to perfection, I could immediately put in my first program. This would permit me to add two numbers together. Marvelous! So I plugged in the two boards, carefully remembering to turn off the computer when I did so. I turned on the power switch. Why are all the little LEDs on and glowing so brightly? They weren't supposed to be on. And – horrors – was that a little curl of smoke coming up from the front panel? With the well known scent of fried IC in my nose I shut off the power. Now where was the smoke coming from? Everything looks OK, maybe just my imagination and perhaps in computers ICs always smell that way. A hollow hope. I turned it on again and saw where the smoke was coming from, a little hole in the center of an IC on the front panel. Not a good sign, I decided. I tore apart the front panel, with the power off, at least I was doing something right. After 20 years I am learning to avoid getting shocked most of the time. Aha, I thought, perhaps there is a short somewhere. Smoke and shorts generally go together, I have learned. I traced the chip in the circuit and found it was a Tri-state buffer fed from the CPU board. So I pulled the CPU board and checked one of the leads going from it to the smokey chip. Hmmm, it's shorted to the 8 volts – and the 15 volts! Not a good sign. That meant everything God and the chip manufacturer meant to get 5 volts of signal was getting all sorts of more enthusiastic information as well. So I checked one of the other leads going to that chip (6 all told) and all showed the same thing. In fact it seemed just about every contact on the CPU board was shorted to every other contact. Assuming MITS hadn't meant it to be that way I concluded I must be the world's champion solder bridge maker. Out with the magnifying glass, no solder bridges to be seen. But what was this along the bottom edge of the CPU board, connecting all the contacts together? A nice hair-thin streak of metal along the plug-in edge of the board. So I filed it off, changed the bad IC, and, forlorn hope that it was, fired up the computer again. Still all the pretty LEDs lit up. I checked the schematic and found that about 15 chips could get zapped by the contacts that were shorted. The one chip I changed was a nasty job and I didn't have most of the other chips anyway. So back in the box and back to MITS for repair.

In the Interim . . .

While waiting for it to come back I decided I had better get on the ball and come up with some sort of terminal to enable me to converse with the computer. As I mentioned earlier, a Teletype didn't turn me on so I gave thought to some kind of TV terminal. I thought about the Southwest TV typewriter but then I'd need either a modulator or a video input added to my TV. I didn't think my Sony color TV would work too well, what with convergence error fringing around the letters, and besides, it was working so well I hated to tear into it. Another hard learned rule — if it works don't mess with it. Besides, if I messed it up how could I watch the Star Trek reruns? Also, it had no power transformer so I'd need an isolation transformer or an opto-isolater to use it. Besides, the TV typewriter writes one page at a time, instead of scrolling up as you write. Also only sixteen 34-character lines fill the screen. Worst of all, no cabinet for it and I am the world's worst cabinet builder. And I really didn't need another hodgepodge of wires and circuit boards in a neat little pile, as most of my projects end up resembling. So I started checking around for a regular used CRT computer terminal. No luck! Remember, this is late 1975. So I decided on a new one. I found an Infoton terminal which is a regular commercial CRT computer terminal with RS-232C interface built in, handsome case, and it has twenty-four 80-character lines on a full screen. It also scrolls up as you type lines into it. Sounded good so I ordered one, and an RS-232C interface from MITS for the computer. The RS-232 interface converts from the 8-bit parallel bus lines of the computer to a serial output to the terminal. The signal levels are also changed. Other standards of interface such as parallel or serial TTL level data are also available but this terminal used RS-232. This is a common interface in commercial equipment. The terminal I chose was a good one as it turned out. However, if I had it to do over again I'd get both upper and lower case, mine has only upper case.

I amused myself by typing on the CRT terminal until I should get my computer back. That lasted about 10 minutes. I also had the other two 4K memory boards to build so I got to work on them. Finally, just when I got ready to go on vacation the beginning of February I found a note from UPS in my mailbox that an $8 COD package was waiting for me. I called them that evening and told them to deliver it to a neighbor to whom I had given the $8. Then I went on vacation fully expecting to have a repaired operational computer when I got back. Ha!

I returned from vacation. No package and UPS had no record of it. After a lot of letters and phone calls I finally located it back at MITS. I sent MITS $8 to ship it prepaid and about three weeks later it finally came again. This time I managed to get my hands on it before it went astray.

Finally!

I unpacked it, set it on the table and plugged it in. All the lites on again? Even for me this was a bit much. Aha, I mumbled, mayhaps a board is loose. I took off the top cover — an act to be repeated so many times I quit bolting it down long ago — and pushed down on all the boards. Seems like maybe one or two were a bit loose. I turned it back on again and, miracle of miracles, it had all the right lites on! At this unexpected turn of events I went to the icebox and got a cold 807. I looked up the machine language program for adding two numbers together and, would you believe, 2+2 came out 4 and 3+3 came out 6. A quick check with the HP-35 showed both answers to be accurate. I now obviously had this computer business, as well as machine language, by the tail. It could only be down hill from here.

I dug out a program from MITS called Kill the Bit and it worked! The lite flashed back and forth among the address lites just as it should and I soon mastered flipping a switch at the right time to kill it. Ah, to think computers were supposed to be complicated — nothing to it. Tonight the computer, tomorrow the world!

I unpacked it, set it on the table, and plugged it in. All the lites on again? Even for me this was a bit much . . .

More Debugging (and Making Bugs)

I plugged in the two memory boards I had built while waiting for the computer to come back to check them.

On one the memory protect lite couldn't be turned off, meaning I couldn't write into it. The other had a bad case of forgetfulness. Well, things were back to normal.

I decided to attack the protect problem first. I looked at the memory board that had been to MITS and saw that a capacitor had been added and an IC pin lifted and connected to an adjacent pin. I looked at the schematic and saw that this was the protect/unprotect flip-flop. I modified my other two boards the same way and found the protect problem was cured.

I recalled reading in the last *MITS Computer Notes* newsletter that some 4K boards required a resistor to be decreased in value to make the refresh circuitry work right. I tried the change and the board now worked for machine language programs.

About this time I finally got the back-ordered expander board from MITS. I put it together, all five hundred solder connections. Before I connected it to the original expander board, another 100 connections, I checked from pin to adjacent pin with my ohmmeter. No shorts so I connected it to the computer. Fired it up after plugging in the memory board that had been to MITS for repair with the computer. More pretty but improper lite displays and a now familiar odor. Quick, turn off the power! Pull the boards and get out the ohmmeter again. Hmmm, a short from one contact to the 15 volt bus. Guess I must have been too generous with the solder. I pulled both expander boards, only breaking a few wires in the process. The short finally turned up on the old expander board where I had attached the new one. Guess I should have checked that too. I wished I could learn to stop shorting things to the 15 volt bus. A simple short between data lines would have done just as well and would not have fried the IC. I promised to keep that in mind in the future. Checked the schematic and found the most likely IC to have been zapped. Miracle of miracles, the tech at MITS for some reason had put in three IC sockets and my selected IC resided in one of them. (I later found out these were the most likely three chips to go west on the front panel so the sockets were routinely installed when they were changed.) And I had a spare too! Well, almost a spare, mine was a 7402 and the original was a 74L02. Well, that would have to do. Not my sort of luck at all. Probably the computer was just setting me up with false hopes before it came up with the ultimate problem.

I cleared the solder bridge, bolted the expander boards back down, and resoldered the wires I had broken off.

Put the memory board back in and it worked again! The rotating bit was rotating again. It just couldn't have been that simple. I shuddered to think of the new problems the computer had in store for me.

Into the Land of Machine-Language Programming

I was still not too sure about how the memories were working so I decided to write a program to test all the memory locations. Seemed simple enough, just write 377 octal (11 111 111 binary) into each location. Then read all the locations back and see if they all had 377 in them. The 377 would put data in every bit in every memory. (Later knowledge shows I had a lot to learn about testing memories.) Well, I dug out the MITS instructions on machine language and got thoroughly snowed. Then I bought *Bugbook III* and read it. Now I understood a little bit about machine language programming and the MITS instructions were much clearer.

So I started to put together a machine language program. Just write 377 into each memory address. Hmmm, that means I'll have to figure out how to tell when the last address is filled, otherwise the computer would write into memory all night. OK, write a test for last address routine, to be used each time the address is incremented. That means I'll need a special program to write into the last address. Not hard, just a few more instructions. Now to check the memory to see if it accepted all the 377s. Not hard, I'll just pull each word out of memory and check it against the word I wrote in, 377, to be sure it is correct. If not, I'll have it put a special word, 252 octal, every other data lite lit, into a memory address I select. Then after the computer run it would be easy to see if any errors were found. The last address problem was harder for readout because of the error routine but I finally figured out a way to do it, with a lot of added instructions. So far so good. Let's run the routine, telling the computer to stop and light the halt light when all the 4000 some odd memory locations have been read into and checked.

Now I knew computers were fast but I'll swear that I had no sooner switched it to run when the halt light went on. Now that's fast! I checked for the data indicating an error and found a 252. That meant I had one or more errors. Hmmmm. But where? Then I got really ambitious and decided to write a new memory test program to write out on the CRT which memory address contained the error. Incidently, during this time I was learning a great deal about machine language programming, although that's not quite what I had started out to do.

This new program became a lot more complicated than I had anticipated but I learned about stacks and calls and pushes and pops and all sorts of things I hadn't needed for the first program. I also got a brilliant idea, for me, about last address testing. Why write a special routine to write into and read the last address after it is recognized when all I had to do was indicate the last address as one beyond the last address I really wanted and use the regular routine to load and read the last address. Going one step further, why not have the computer add one to the real last address when testing. I was pretty proud of that idea but I suspect it wasn't original with me.

Now I knew computers were fast but I'll swear I had no sooner switched it to run when the halt light went on.

Well, to make a long story short, after spending several evenings writing the program and about an hour loading it into the computer, it didn't work. It wasn't a total loss, because I sure learned a lot about machine language programming, ins and outs to and from the terminal, data ready tests, rotating to check data, and many other things.

Then I decided, well, maybe it's my program that's wrong and the memories are really OK. So I decided to go ahead and try to load BASIC.

Program Loading Woes

Ah BASIC. For those who are just getting interested in computers BASIC permits you to type programs into the computer in a language resembling English. You use words like IF, THEN, FOR, PRINT, etc. Also some I never heard of like DIM, SQR, SGN, ABS, RND, etc. But it's really not hard after a little reading about it. Much easier than machine language with all its numbers and much faster to type in letters than flipping switches.

In the Altair manual it tells all about how to load BASIC. I followed the instructions and played the BASIC tape into the computer. At the end of the tape the terminal is supposed to type out "Memory size?". In my case it printed ??P AXYZ or somesuch. Well, about what I suspected. I tried a few more times and got the same thing. I researched a few of the MITS newsletters and found a different ACR setup routine. Remember, the ACR is the hardware interface between the computer and the tape machine.

First it said to check the tape recorder for proper speed with the data on the flip side of the BASIC tape. So I took the recorder to work and set the speed on a frequency counter we had there. Tried it again when I got home, same garbage printout when the tape ended. Went through the ACR setup in the MITS newsletter, same thing. Tried an echo routine given in the MITS manual and it worked fine. This means you put this program into the computer and type on the CRT with it set for duplex. This means the signal is going from the keyboard to the computer, being digested by the computer, and then sent back to the CRT terminal where it is printed. I decided that eliminated the CPU and I/O board, leaving only the memories. I called MITS and the tech there gave me a program to send letters from the switches on the computer to the terminal to make letters for another check for proper operation. I loaded the program and sent letters to the terminal. At least flipping switches changed the letters so I concluded all was OK. He also gave me several component changes on the memory board to try which didn't help either. So I decided to send two of the memory boards back to MITS, keep one to play with, and order two new assembled static memory boards from Processor Technology. I figured maybe by buying static memory boards I could avoid the troublesome 74123 one shots in the dynamic memory's refresh circuitry.

While waiting for the new or repaired boards I decided to play with the computer-to terminal information some more. I dreamed up a very simple program to send letters from the terminal to light the data lites on the computer's front panel. Hmmmm. Seems the data the computer is showing is different from what I'm sending. I tried it critically the other direction and found that the computer switches were making letters, but the wrong ones. I finally figured out that I should be sending 7 bits from the terminal to the computer, but the CRT I/O board was jumpered for 8 bits. That took an evening to figure out. I still don't know why the echo program worked. I changed the jumper to 7 bits and started getting the right data to the computer. After a lot of head-scratching I found that I was using the wrong switches to send data from the computer to the terminal. The data was right, I just wasn't sending what I thought I was. The data switches and sense switches (the sense switches were the ones I was using) were in a long row and I had misread the front panel and was using switch patterns one switch to the left. That took another two evenings to figure out. So now I was beginning to suspect pilot error rather than memory faults. I then received the Processor Technology memory boards (very fast service) and plugged them in.

Processor Technology thoughtfully included a machine language program for memory testing which I tried on their two memory boards and the remaining MITS board. They all passed with flying colors. I suspected now that the two memory boards I had sent back to MITS were really OK. The MITS memory problems were undoubtedly caused by my mis-switching and jumper error.

Into the Land of BASIC

I loaded BASIC again, using a MITS board and one of the new boards and it loaded fine. I put the third 4K board in and had my first experience with BASIC.

So now I have BASIC loaded. The first thing I load is a simple-minded program designed for young users where the computer selects a number between 1 and 100 and you have to guess what it is. The computer tells you if you are too high or too low and you guess again and so on until you guess the number. I played this with the computer for a half hour and really enjoyed it. Shows what waiting five or six months to run a BASIC program can do to you.

Next tried a slot machine program and won a bundle from the computer. Then I loaded a simple Blackjack program and lost $20,000 to the computer. Sat up until 3 am playing Blackjack with a machine! Using imaginary money! Incidentally, I got those games from a book called *101 BASIC Computer Games* put out by Digital Equipment Corporation ($7.50).

Then I tried writing a BASIC program, running it to be sure it was OK, then I used BASIC to load it onto cassette tape, then reloaded it into the computer and it worked. The right program came back and I could run it again. I loaded a couple of simple programs and then typed (for about 2 hours) a program from the book for an elaborate Blackjack program (splitting pairs, insurance, etc.) but it wouldn't fly. The computer tells me it's out of memory but a command to tell me how much memory I had left said I still had over 400 words. Now I know that it was telling me it was out of string space. A one line statement would have cured the problem. Ah, hindsight. Unfortunately all BASIC languages are not exactly the same. Sometimes you have to change a statement slightly, or add or delete a statement. It doesn't take very long to know what to change. I found that most programs will run OK without changes. After you read a book or two about BASIC you learn what those mysterious commands mean and can devise a way to do the same thing using commands you have in your version. MITS BASIC also has error printouts which can be helpful in finding the source of the problem.

At any rate, I then loaded a program for Lunar Lander, and crashed every time. I would make a poor astronaut. In this program you get a readout of how fast you are falling, how far you are from the surface, and how much fuel you have remaining every second from a start at 500 feet above the surface. You input the amount of fuel you want to use for that second. You have a limited amount of fuel and I generally ran out before I touched down. Crash!

As you can see with 101 games in that book I have barely scratched the surface and there are other game books available as well.

Some games like Star Trek require quite a bit of memory, the shortest fairly complete version I've seen uses 4K for the program. That's one reason why I ended up with 20K. If you stay with machine language you can be very happy with 4K or less. Or with 8K BASIC and running medium sized programs 12K is plenty, even 8K will get you by on the shorter programs.

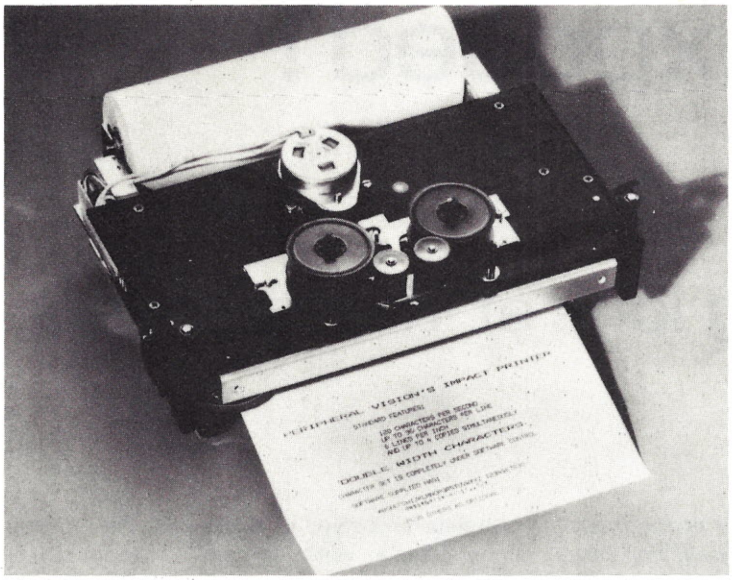
Once you know some BASIC and have run out of interesting game programs try getting a book called *Problems For Computer Solution.* It is full of ideas for interesting problems of all sorts to turn into BASIC programs to solve. It is a great help in learning to use BASIC better. It will keep me out of mischief for a long long time.

The Little 'ol Problem Maker, Me

What I am trying to say is that computers aren't that hard to learn to use and program. Most of the problems I had were brought on by myself. They generally weren't in the equipment but were in my hasty decisions or failure to really read the instructions or follow the factory's instructions properly. Besides that, I learned a great deal about computers along the way, as well as writing software (and I still have a long long way to go). For me writing programs has turned out to be my favorite part of the computer hobby. I bought my computer almost 18 months ago now, and I have done very little but work with one aspect or another of my computer in my spare time. I've tried a lot of hobbies, but this has been the best by far. At least it seems that way now. However, I can remember the time many years ago when I first got into ham radio. I suppose I felt the same way then.

If this gets printed, and the kindly editor is agreeable, I will write about my further adventures in computers. Purchasing a disk memory and trying to start a business with my computer will follow. If you think I had problems so far . . . ■

Peripheral Vision impacts your computer.



WITH A FULL-SIZE, LOW-COST IMPACT PRINTER.

Until now, the hobbyist and small businessman have had one major problem in assembling a reasonably price microprocessor system with the capabilities found in the more costly computers. It was impossible to find a high-quality, high-output printer for hard copy needs at an affordable price.

Peripheral Vision has come up with a solution.

We are offering a full-size *impact* printer designed for microprocessors—and it comes with a mini price. Kit prices start as low as $495 for the printer and interface card. And that won't impact your pocketbook.

Peripheral Vision's printer is loaded with capabilities. Take a look:

- It's fast—120 characters per second
- 96 characters per line, 12 characters per inch horizontal, 6 lines per inch
- Makes up to 4 copies simultaneously
- Character set and pitch variable under software control
- 5 x 7 character matrix
- Ribbon has built-in re-inkers for a life of 10,000,000 characters
- Paper can be either a standard 8½-inch roll, fanfold or cut page
- Interfaces to 8-bit parallel ports

Just remember, Peripheral Vision is committed to helping you get along with your computer. The new printer we are offering is another example. It is high quality, low in cost and will definitely impact your system.

Write or call now to find out how to impact your computer.

PERIPHERAL VISION INC

P.O. Box 6267 Denver, Colorado 80206 (303) 777-4292

Peter A. Stark
PO Box 209
Mt. Kisco, NY 10549

Prototyping Systems Exposed!

...a revealing look at the Intercept Jr.

If you've been thinking about going the route of the Intersil IM1600 (particularly the Intercept Jr.) then you'll find Pete's article very interesting. On the other hand, you'll find it worthwhile reading simply because of the interesting points about the system which are brought out. When I first saw the Intercept Jr. and its flashlight "power supply" I had a little trouble taking it seriously ... then I saw an Intersil demonstration which turned me into somewhat of a believer. – John.

When the first microprocessors came out, the IC manufacturers at first tried to market them the same way they sell other ICs - put out a short spec sheet, put a full color ad in a few engineering magazines, and wait for the orders to pour in.

They soon discovered that you can't sell microprocessors like other ICs. In addition to the spec sheets they found they also had to provide detailed hardware manuals, programming textbooks, sample programs, troubleshooting and debugging programs, and even whole operating systems. In other words, they had to train their customers to use their devices before they could convince them to buy.

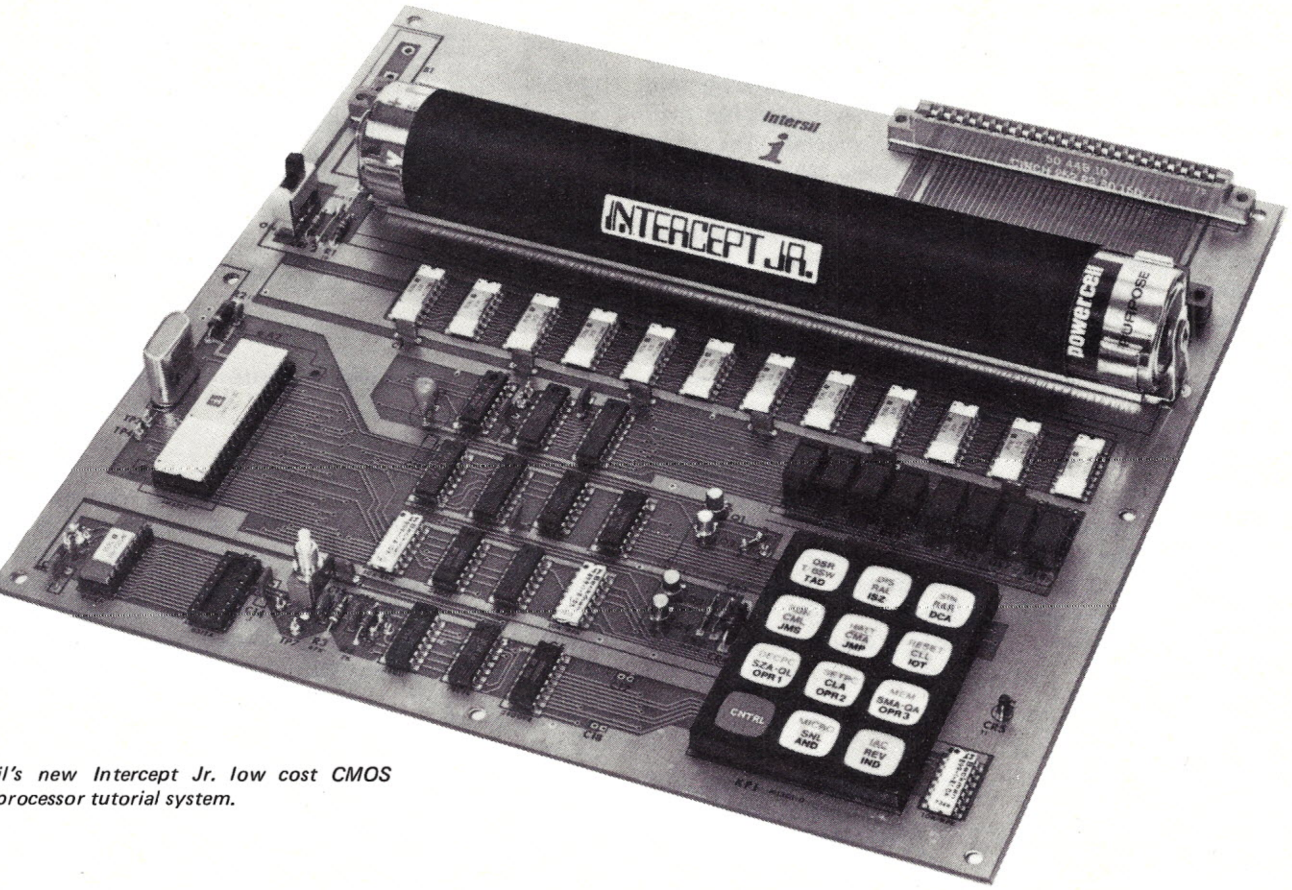
As a result, each microprocessor IC manufacturer today puts out one or more complete microcomputers using his ICs. These units are not intended for the hobbyist market and are often called *trainers* or *prototyping systems* rather than computers. They are aimed at engineers in other companies who want to learn about and experiment with a microprocessor so they can design it into some other piece of equipment such as a cash register or some test instrument. In some cases these units are so simple that they can only be used to learn about a particular microprocessor; in other cases they are big enough that they can be connected to other equipment to make a complete prototype of something like a working cash register.

These trainers or prototyping systems differ from the hobbyist computers like an Altair or Imsai in several ways. Since they are designed for engineers, the manuals which come with them are often somewhat terse and full of technical terms. They often come without a cabinet or control panel. They do not have a common bus structure to which you can easily connect other devices. On the other hand, some are really cheap. For example, National Semiconductor makes a unit which has their SC/MP microprocessor, ROM and RAM memory and a few other components on a printed circuit board about 5 inches square and costs $99. A keyboard and LED display is available separately in a plug-in package that looks like a calculator and costs $95. Put the system together and you have a working, though small, microcomputer for under $200.

The Intersil Intercept Jr. Tutorial System is another such unit, though slightly more expensive at $280. And there are others. Each of these systems can be expanded to a large enough system to make any hobbyist happy, though it's obviously not as easy as just going out and buying a board to fit your Altair bus.

Since the Intersil has the advantage of having the IM6100 microprocessor which takes the same programs as a Digital Equipment Corporation PDP-8e, the Intercept Jr. trainer has a great attraction to anyone who has any experience with a PDP-8e or who has access to the tremendous variety of programs available for it. This article describes this trainer and tells you what you get for your $280.

Fig. 1 is a block diagram of the Intersil trainer. As you can see, it consists simply of the IM6100 microprocessor tied to a data, address and control bus, with everything else tied to the bus. In this



Intersil's new Intercept Jr. low cost CMOS microprocessor tutorial system.

case, the bus consists of 36 wires running up and down the printed circuit board, which carry instructions and data as well as control signals. Connected to this bus is a Read Only Memory (ROM), a Random Access Memory (RAM), a keyboard and eight-digit LED display, and assorted control circuits. Besides the microprocessor and memory ICs, the whole board has only 12 other control ICs.

The control and data bus is also connected to three 44-pin card connectors into which you can plug your own peripheral equipment or more memory. Intersil makes three plug-in boards which may be added - a 1K RAM module (very expensive at $145), a 2K PROM module (also expensive at $75 because it comes without the PROM ICs), and a Teletype interface using a UART which can drive either a Teletype current loop or an RS-232C interface (a bit more reasonable at $82). The Intersil RAM module is expensive because it uses CMOS memory ICs; memory can be added cheaply by using the more popular 21L02s or other common memory ICs.

Because of the three 44-pin connectors, adding your own modules is fairly easy. Radio Shack, for example, has several small prototyping boards with 22-pin connector fingers at one end; these are available for $2.95 each and are a perfect fit for smaller interfaces. You can also use one of these boards as a plug, with the connections brought out through a cable to external memory boards or even to an external mother board like that of an Altair or Imsai.

Let's take a more detailed look at each component of the overall system.

IM6100 Microprocessor and Bus

The heart of the system is, of course, the Intersil IM6100 microprocessor. There is enough to this IC that it requires an entire article to describe it, so I will not give details on that here. As shown in Fig. 2, the bus used for data and control signals simply consists of 36 wires coming out of the IM6100 and running over to the 44-pin card connectors. (Connections to other components are not shown in Fig. 2.) Six of the IM6100 input lines are not used elsewhere on the board and so they are connected to the positive supply voltage Vcc through the six 10k resistors shown in the lower right corner. In addition to the 36 bus connections, the card connectors also are connected to ground and Vcc, as well as four additional signals coming from the memory and control logic.

Clock

As shown in Fig. 2, the clock for this microprocessor consists of one part - the crystal connected to pins 14 and 15. The IM6100 is a CMOS static processor which does not need complex clock circuits; it has a built-in oscillator whose frequency is controlled externally by that crystal. It's that simple!

Depending on the particular version of the IM6100, the crystal frequency can be from almost dc up to as high as 8 MHz. In general, CMOS logic operates faster at higher supply voltages, and so the operating frequency depends on the supply voltage required by that IC. The *standard* IM6100 will work up to 4 MHz with a 5-volt supply; at this speed it can complete a typical instruction such as an add in 5 microseconds. A premium version, the IM6100A, can run at 8 MHz and 10 volts, while a cheaper IM6100C will run at only 3.3 MHz at 5 volts. The Intercept Jr. system comes with a 2.46 MHz crystal and an IM6100C; the slower crystal is used to permit reliable operation even if the power supply voltage drops down to 4.5 volts. Obviously, with such a simple clock, you can easily change the speed to fit the power supply and also to fit the speed of the memory used. For troubleshooting purposes it is even possible to provide single clock pulses from a switch and step the processor through one step at a time. Alternatively, the clock speed could be made really slow (such as 100 Hz or 1000 Hz) so that any cheap oscilloscope could be used to observe waveforms.

Power Supply

The power supply isn't shown in any of the diagrams because it is so simple - four D cells! They mount right on the circuit board and run the whole system for weeks, maybe even months.

When operating, the entire system (not including the LED displays) takes less than 20 *milliamperes*. The eight LEDs alone take almost 200 ma. As a rough guess, the four batteries should last ten to twenty hours of continuous use if the LEDs are used continuously. If the LEDs are turned off when not needed (and this is easily done under program control), the batteries should last for several weeks. Since the processor runs at a slow speed, battery voltage can drop from 6 volts down to 4.5 volts and the system will still run. (When I finally replace the battery supply with a power supply, I plan to increase the crystal frequency.)

The low battery drain shows up in a different way too. When the machine is stopped, the CMOS RAM memory ICs take a total of about *5 microamperes*! This is such low power that the RAM memory is connected right to the battery, bypassing the power switch. In other words, even with the power turned off the memory is still on and powered. Hence

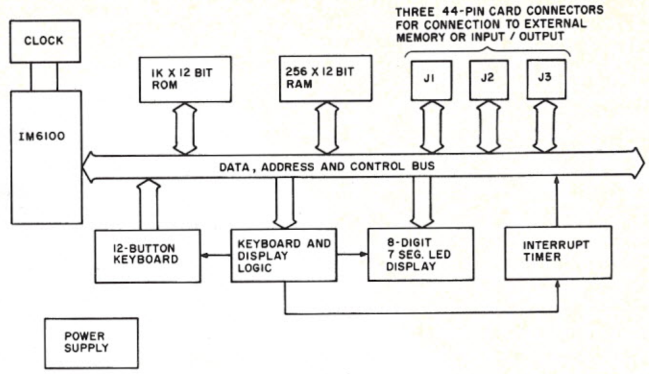


Fig. 1. Block diagram of Intersil "Intercept Jr."

The schematics associated with Intercept Jr. are reprinted with permission from Intersil.

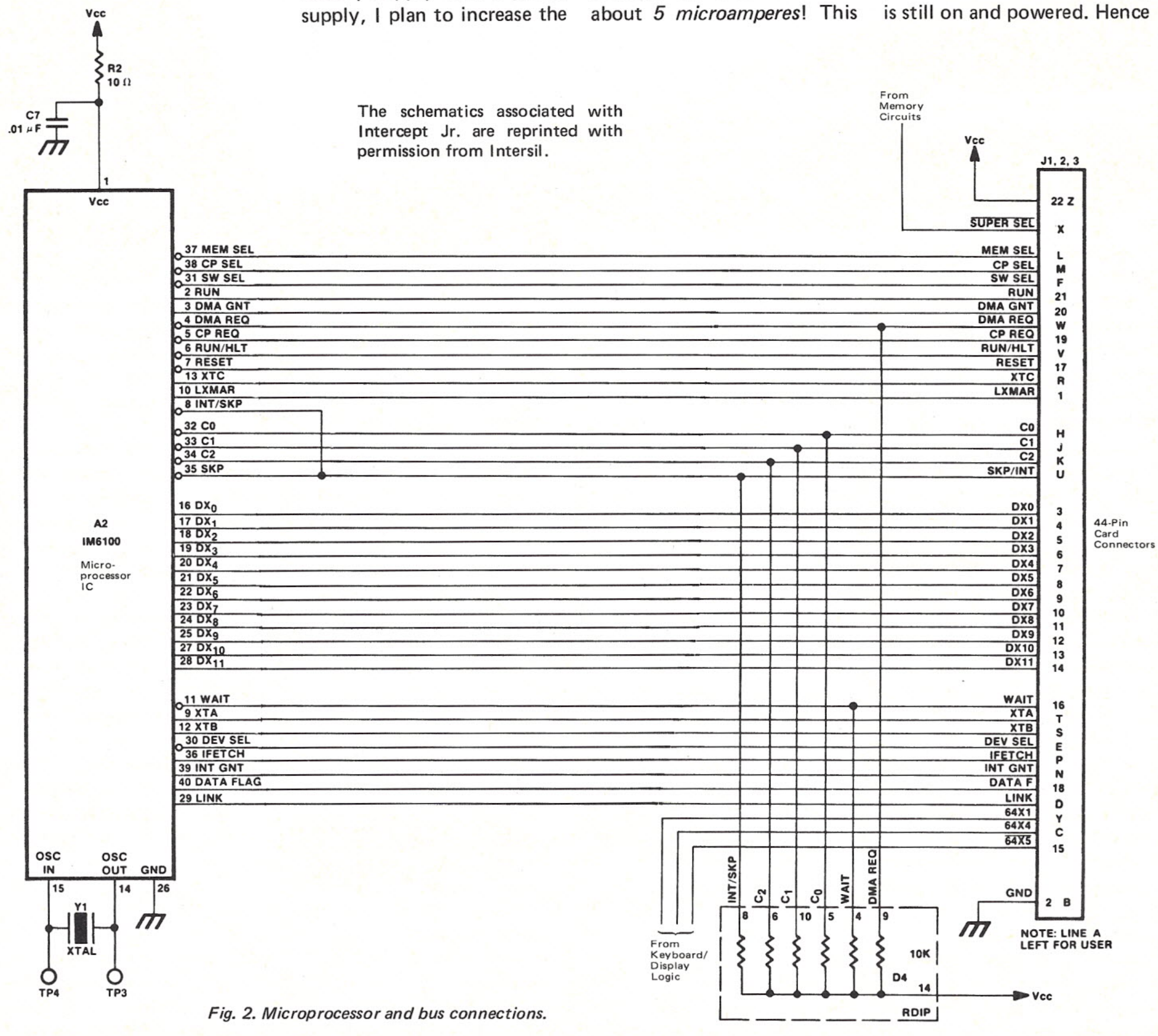


Fig. 2. Microprocessor and bus connections.

you can leave a program in memory, turn the power off, and return a week or a month later and have the program still there. At a 5 microampere drain, the batteries will last for their full shelf life of a year or two.

Read Only Memory (ROM)

The Read-Only-Memory, shown in Fig. 3, consists of just one IC, an Intersil 6312, and here is the source of one difficulty. This is a special IC, which is mask-programmed at the factory and for which no programmable replacement exists. Although there is a spare socket on the board for a second ROM, it is unlikely that you will ever get one unless Intersil comes out with a preprogrammed ROM in the future for some specific purpose.

This ROM has 1024 words (1K) of 12-bit memory. As supplied with the computer, it contains a program called (somewhat incorrectly) a microinterpreter. This is a program occupying memory locations 6000 through 7777, whose function is to monitor the keyboard, drive the LED display, and in general perform the functions normally done by a control panel on the PDP-8e. This system has no control panel in the traditional sense; instead, the program in this ROM in conjunction with the keyboard and the LED display allows the loading and displaying of data and instructions in memory, starting and stopping programs, and also punching or reading programs on paper tape via a Teletype. This program also allows us to step through a program one instruction at a time and observe the results.

At this point it's time for a slight detour to explain the term *software control panel*. Minicomputers and large computers traditionally have a hardware control panel through which it is possible to enter numbers or read out numbers from the various registers in the computer. But such a control panel requires connections directly to these registers; in a microprocessor these registers and other control circuits are buried inside the IC and are not accessible from the outside. So it becomes very hard to build a real control panel out of hardware. On the other hand, it is possible to get to these registers with a program. Most recent microcomputers do not have a hardware control panel at all, but instead have a software control panel program on a ROM which does all the functions. In some microprocessors this ROM was put in as an afterthought; in the Intersil IM6100 the microprocessor IC was already designed with this in mind.

The IM6100 was designed to have two completely separate memories - a main memory for the user and a control panel memory for the software control ROM and a small amount of RAM memory to go with it. By keeping the two memories separate the two will not interact and cause problems. In this way it is also possible to use the maximum amount of user memory without taking any away for control panel programs.

Pins 37 and 38 on the IM6100 specify which of the two memories the processor wants to access. In the *user mode* the processor puts a pulse on pin 37, called MEMory SELect, each time it

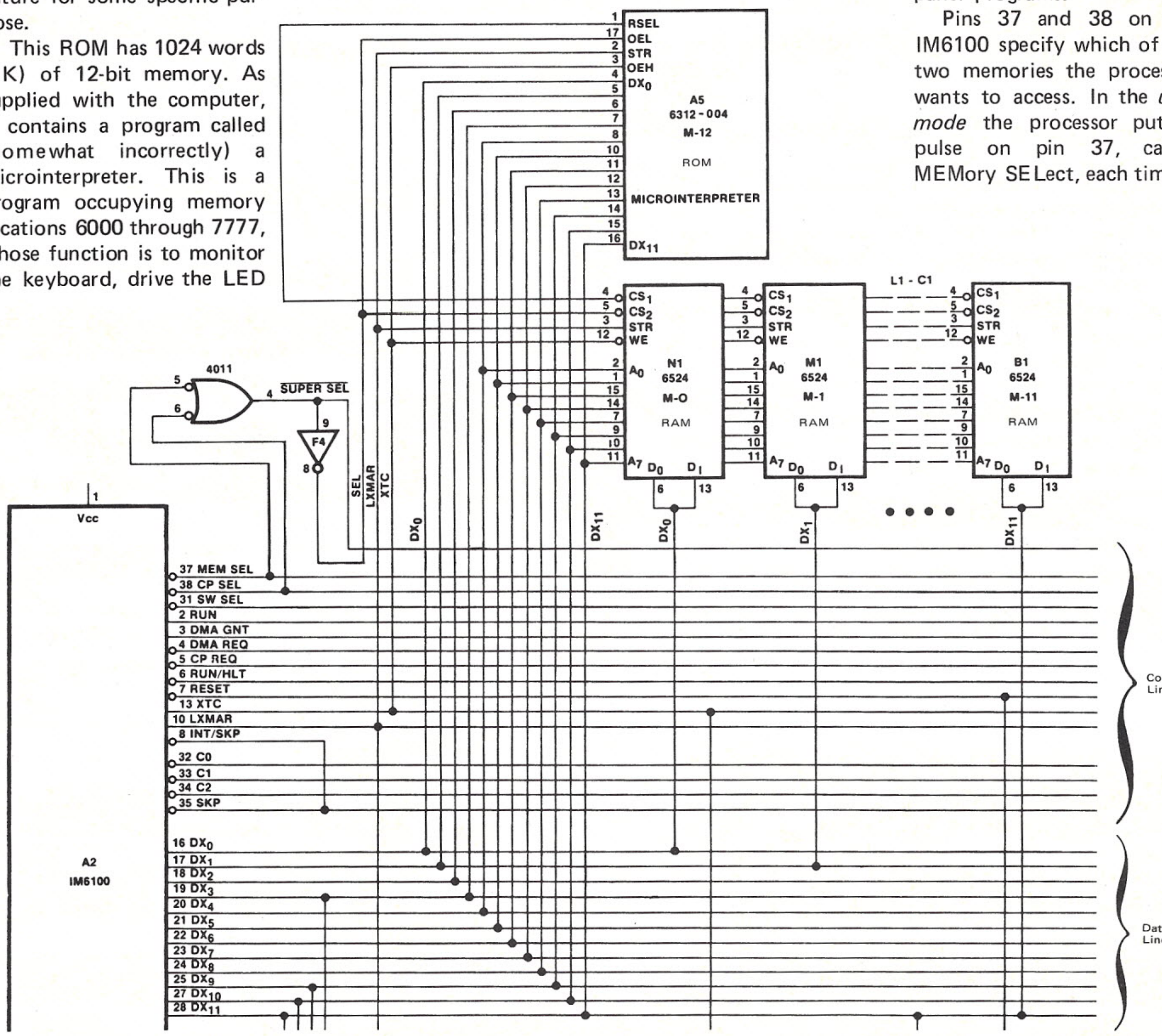
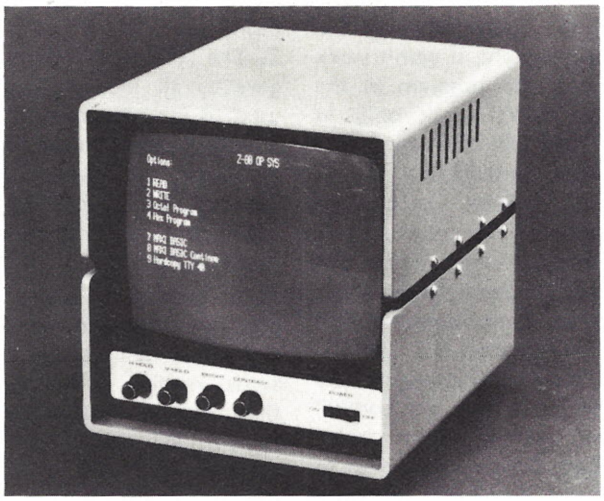
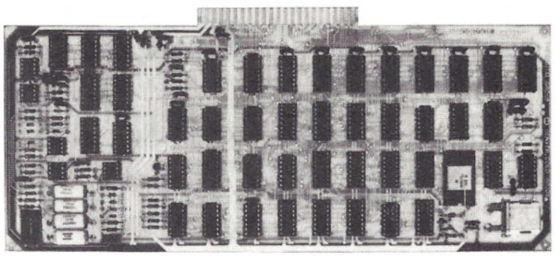


Fig. 3. ROM and RAM.

The Digital Group adds character(s).



64, to be exact.



The Digital Group's computer systems have a lot of character already. Just one quick look at any of our products in their unique custom cabinets confirms that. But we believe it never hurts to add a bit more.

So, the Digital Group has added character in a big way to give an added dimension to the operation of our video-based computer systems. We are pleased to announce our new TV readout with a 64-character line. It will give your system a great deal more capability. Give it more character, if you will.

Here are the specifics on the Digital Group TV Readout and Audio Cassette Interface:

1024 Character TV Readout

- 64 characters horizontal by 16 lines
- 7x9 character matrix (effectively 7x12 due to character shifting)
- 1K on-board RAM for buffer storage—requires no main memory—completely independent
- 128 character ASCII
  - Upper case alpha
  - Lower case alpha with base line extenders (g, j, p, y)
  - Numbers and extended math symbols
  - Greek alphabet
- Software driven cursor—forward and backward
- Compatible with most microprocessors; Interfaces with 1 8-bit parallel output port
- Timebase may be driven with an external timebase (may be synchronized to TV camera, TV set, etc.)
- Readout timebase available at connector (can be used for graphic driver, etc.)
- White characters on black, and/or black on white; software selectable
- Plugs into standard dual 22-pin TVC connector on Digital Group Systems

Improved Audio Cassette Interface:

- Reliable FSK recording technique
- Uses standard unmodified audio cassette recorder
- Write cassette system uses a digitally synthesized frequency shift system, derived from TV system's master crystal oscillator
- Read cassette system easily aligned using the write system as an alignment aid.
- Runs at 1100 baud (100 characters/second)—loads 16K in 3 minutes

512 TVC to 1024 TVC Upgrade Kit:

As always, when the Digital Group extends the capabilities of our systems, it doesn't mean obsolescence for any products. We are offering an upgrade kit for present Digital Group system owners who wish to go to the longer line length. This kit uses most of the IC's from our TVC-F readout. No unsoldering is required; all new sockets, capacitors, resistors, PC board and other necessary parts are supplied.

Prices:

TVC-64—Full 64-character TV Readout & Audio Cassette Interface:

Kit — $140 Assembled — $205

TVC-64UPG—Upgrade kit from TVC-F:

Kit — $65

If you already own a Digital Group system, our 64-character line will definitely enhance its operation. If you're just looking, you might want to keep in mind that the Digital Group has a lot of characters.

Write or call now for details on our new 64-character TV readout and all our other exciting products.

the digital group INC

box 6528 denver, colorado 80206 (303) 777-7133

wants to write into or read from user memory. When performing control panel programs, it puts a pulse on pin 38, Control Panel SELect or CP SEL each time it wants to write or read from control panel memory. (In order to perform the control panel programs, it is necessary to have a small amount of RAM into which various constants can be written; hence the control panel memory needs not only a ROM but also some RAM.)

This is where Intersil took a shortcut in designing this prototyping system. Although the IM6100 has provisions for *two* different memories, the Intercept Jr. system only has *one memory*: The RAM is used for both user programs and control panel programs. They did this by putting in a 4011 CMOS gate, connected to pins 37 and 38 in such a way that a SUPER SELect signal is generated for either user memory or control panel memory requests; the resulting SELect signal is sent to both the ROM and the RAM. With the memory supplied, locations 0000 through 0377 (in octal) are the RAM memory, while 6000 through 7777 (octal) are the ROM. Of these, 0140 through 0177 (and 6000 through 7777) are used by the control panel programs, and the rest is available for the user. Not much.

Now, this is where the problem comes. You can add more memory, either by buying the expensive Intersil CMOS RAM modules, or by building your own from standard NMOS ICs. But since the only unused addresses are 0400 through 5777, you can only add slightly over 2K. It's hard to add just a part of a K, so it is only reasonable to add 2K. This is still not much.

Ah, you say, why not break up the control panel/user memory into two, so that you can have the full 4K for user programs, and put the control panel routines into their own memory? Technically, this could be done by removing that 4011 gate connected to pins 37 and 38, so that the MEM SEL signal does not go to the built-in ROM and RAM. Unfortunately, it won't work because the program in the ROM won't work only in control panel memory. It has several instructions in it which reach into user memory when they shouldn't. And because the ROM is mask-programmed and not interchangeable with any programmable ROMs, you can't even change it. So breaking up user and control panel memory is a major project, because you have to tear out this ROM and carefully interface another one into the system. (Hopefully Intersil will recognize this problem and come out with a new one.)

Now back from the detour, and back to Fig. 3. Twelve of the lines in the bus coming from the microprocessor are called DXO through DX11. These twelve lines carry addresses, data, and input and output commands and data. They are like a major highway which is shared, at different times, by different kinds of numbers. These twelve lines connect to the ROM to supply it with memory addresses, and also to get instructions out of the ROM. When the processor uses the memory, the typical sequence of operations goes like this:

1. First, the processor throws out an address on the data lines; this is the address of the memory location which it wants to access. An instant later, it puts a short pulse on the line labelled LXMAR (pin 10), which stands for Load eXternal Memory Address Register. Each of the RAM and ROM ICs has a built-in group of flip-flops called a memory address register which grab the address when they receive the LXMAR signal and save it. (If additional memory is added using other memory ICs, then additional flip-flops may have to be added to duplicate this register; the Intersil memory ICs are special in that they already have this register built in.)

2. The XTC line, which also goes to all memory ICs, is used to indicate whether the processor wants to read from memory or write into memory. In the case of the ROM, this will always indicate a read only.

3. A pulse on either the MEMory SELect or the Control Panel SELect starts the actual reading or writing. In this case, the gate connected to pins 37 and 38 of the IM6100 generates a single SEL pulse which goes to both the ROM and RAM memory ICs. When the ROM gets this pulse it puts the instruction or data in the specified location out onto the twelve-bit data bus and it is sent back to the processor.

One last comment on the ROM IC is that it is internally mask programmed so that it responds only to addresses starting with a 6 or a 7. Thus it contains all locations having addresses of 6000 through 7777.

Random Access Memory (RAM)

The RAM memory is split up among twelve ICs, of which only three are shown in Fig. 3. Each memory IC is organized as 256 one-bit words (see my article on memories in the April, 1977, issue of *Kilobaud* for a discussion of memory organization.) Thus a twelve-bit memory location is spread out over all twelve ICs, one bit in each. Each RAM IC has a DI (Data In) pin and a DO (Data Out) pin, which connects to one of the data lines in the bus, and it stores the one bit appearing on that line.

In keeping with the 256 memory addresses it can store, each RAM IC also takes eight address bits from the bus; bus lines DX4 through DX11 go to address pins A0 through A7 on each. But it takes the full twelve bits to specify an address, and this is where the ROM IC comes in again. (Remember, this is a special IC made just for this application.) The ROM looks at data bus bits DXO through DX3, and each time they are all zeroes, it provides Ram SELect (RSEL) output which goes to pin 4 of all the RAM ICs.

Hence, as you can see, the system needs no other interface ICs to connect the memory directly to the microprocessor. But that is somewhat of an illusion, since these are special memory ICs which contain internal address registers, and the ROM also contains some of the required address circuitry to drive the RAM. If you were to add additional memory to this system you would have to add this circuitry as well.

Keyboard

As shown in Fig. 4, keyboard wiring consists of twelve resistors, twelve key switches, and twelve inverters (only four are shown of each). The resistors normally keep all inverter inputs near +6 volts, but the keyboard is wired so that it can short any input to ground.

The inverters have *three-state outputs*. This means that their outputs have three states — high voltage, low voltage, and open circuit. In the normal state they are open-circuited, and send no signal to the data lines.

Like the PDP-8e, the IM6100 has two instructions designed to input data from a so-called switch register into the computer. On the PDP-8e, the switch register was a group of twelve toggle switches which were wired so that their state could be entered into the arithmetic unit of the computer when the appropriate instruction was given. In the IM6100, this instruction simply puts a pulse on pin 31, called SWitch SELect. This pulse is sent to the twelve inverters and turns on their outputs. And bingo - the state of the

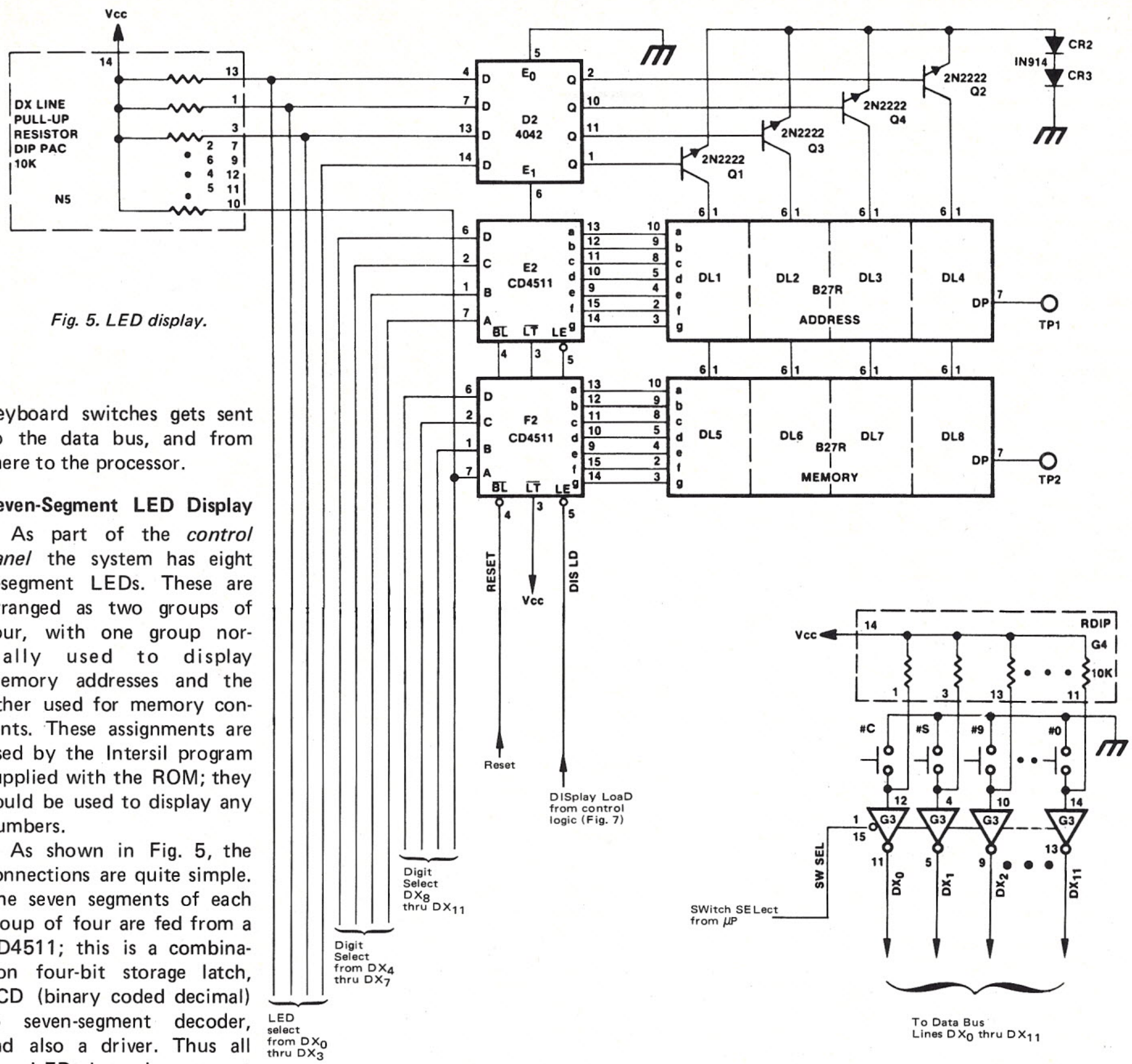


Fig. 5. LED display.

Fig. 4. Keyboard.

keyboard switches gets sent to the data bus, and from there to the processor.

Seven-Segment LED Display

As part of the *control panel* the system has eight 7-segment LEDs. These are arranged as two groups of four, with one group normally used to display memory addresses and the other used for memory contents. These assignments are used by the Intersil program supplied with the ROM; they could be used to display any numbers.

As shown in Fig. 5, the connections are quite simple. The seven segments of each group of four are fed from a CD4511; this is a combination four-bit storage latch, BCD (binary coded decimal) to seven-segment decoder, and also a driver. Thus all four LEDs in each group get the same digit, but the transistors above the LEDs only turn on one of each four LEDs at a time. These transistors in turn are controlled by a 4042 latch. Twelve resistors, shown in the upper left corner, pull the data lines up to about +6 volts when they are not used for anything else.

The operation of this circuit is quite simple. When the microprocessor in response to a program wants to display a pair of digits, it puts a group of twelve bits on the data lines, and a DISplay LoaD signal (from the control logic) enters these into the three latches. Of the twelve bits on the data lines, the first four select one pair of LEDs, the next four provide a digit to the top LED of the pair, and the last four provide a digit to the bottom LED.

Keyboard/Display Logic and Interrupt Timer

The remaining circuitry on the printed circuit board is the control circuitry for the keyboard and display, as well as some of the timing circuits which make the control panel program work. This is special circuitry not very adaptable to other systems, and so I will cover it only very briefly for those who are interested.

These circuits are controlled by the microprocessor by use of special Input Output Transfer (IOT) instructions similar to those done by the PDP-8e. A typical IOT instruction looks like 6032 (in octal). Here, the 6 specifies that this is the IOT instruction, the 03 in the middle specifies one of a number of different devices involved (such as a Teletype keyboard etc.), and the 2 on the end specifies what that device is to do. This arrangement can be used to control up to 63 different input/output devices by choosing the appropriate combination of middle two digits; these middle two digits can be any octal number from 01 to 77, corresponding to 63 different codes. (The 00 combination is used internally in the microprocessor for controlling the interrupt system.)

In the Intercept Jr., the designers took a short-cut, however. The circuitry which controls the keyboard, display, and related circuits does not examine the entire six bits corresponding to this two-digit octal code; it only looks at the first bit. Hence any code which starts with a

1 (which includes all the codes starting with 40, 50, 60, or 70 in octal) is interpreted as applying to the keyboard/display/control panel circuits. Though this cuts down to 31 the number of possible input/output devices you can add to the system, this is not particularly a limitation for the average hobbyist. Since the device code is any number above 40, the instruction manual and drawings specify the code as 4X to indicate that only the first bit is important. (The symbol X usually means that we *don't care* about that digit.)

Fig. 6 shows the circuitry which catches the IOT instruction as it comes out of the microprocessor and decodes it. At the beginning of the IOT instruction the microprocessor feeds out the entire instruction on the data lines just as if it were a memory address. The DX3 bit, which is the first bit of the device code, and bits DX9 through DX11, which comprise the last octal digit of the instruction, go into a 40175 latch. When the LXMAR (load external memory address register) line gets a pulse, the latch quickly stores those four bits and then applies them to the 74C42. Depending on the particular bit combination at its input, this latter IC may ground one of its outputs. If, for example, the instruction is a 64X0, then pin 1 will be grounded and so on. Each of these eight different instructions will therefore ground one of the eight pins. Only five instructions are needed for the control circuits, however, and the unused outputs (64X1, 64X4, and 64X5) are simply brought out to the card connectors for any use you may have for them. This is shown in Fig. 2 and will be used later in this article.

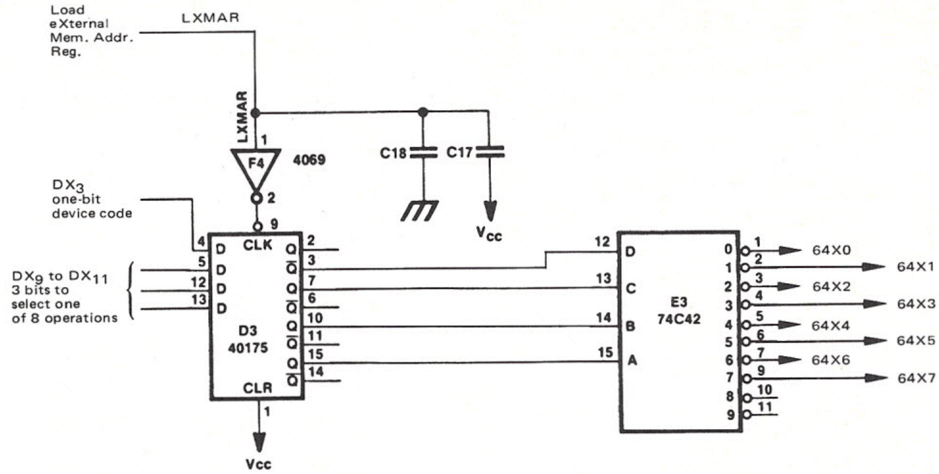


Fig. 6. IOT instruction decoder.

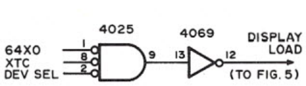


Fig. 7. Display load circuit.

Fig. 7 shows the display load (DIS LD) circuit used to feed the latches in Fig. 5. When a 64X0 instruction is given, this circuit waits until the microprocessor provides an XTC signal and a DEVice SELect signal several microseconds later and then provides a DISplay LoaD signal to the latches. The timing is such that this occurs just as the microprocessor is outputting the twelve bits which tell the latches which digit to light up on which LED.

Fig. 8 shows the interrupt timer and its control. Keep in mind that the control panel of a computer is supposed to be usable even when the computer is halted or when it is busy doing another program. This is achieved with a software control panel by having an interrupt timer which continuously interrupts the microprocessor and forces it to jump from whatever it is doing (or not doing) into the control panel program. In this system this is done by having a timer (in the center of Fig. 8) which provides a sharp pulse output about 100 times each second. This pulse sets the CP Interrupt Flip-Flop, which then generates a CP INTerrupt REQuest pulse to the processor. (There is some additional gating here which resets the flip-flop once the IM6100 acts on the interrupt and also prevents a control panel interrupt if an interrupt is already in process from some other cause.) The interrupt flip-flop can also be activated by a 64X3 instruction, using the 4025 gate shown below it, so that any program can go to the control panel routine if desired.

Under some conditions it may be necessary to disable the interrupt timer. For example, if the microprocessor is in a loop for the purpose of timing out some specific time interval, the timing would go wrong if there were control panel interrupts at random times. A 64X2 instruction can therefore be used to control a Timer Control Flip-flop, turning it on or off depending on the state of the DX11 bit.

Finally, a real control panel must be able to start and stop the computer and to reset all registers. Fig. 9 shows the circuitry which does this. The RESET signal is generated at three possible times: (a) when a 64X6 instruction is done, or (b) when the reset switch is closed, or (c) when power is first applied. The RUN/HLT signal is generated whenever a 64X7 instruction is performed.

This last circuit deserves a bit of explanation. Keep in mind that these circuits are intended to be used by the control panel program in the ROM. The control panel is entered whenever a control panel interrupt is caused by the timer, even if the machine was stopped before the interrupt. Thus the computer might have been stopped in the user mode, yet running a few milliseconds later in the control panel mode. Pulsing the RUN/HLT line at this point would restart the computer when it returns from control panel mode back into user mode. Without this explanation it's a bit difficult to understand how a 64X7 instruction can be performed to start the machine while it is halted.

Interfacing Example

If by now you feel like you have just been taken through the wringer, don't be surprised. After all, in the space of just a few pages you have gone over every single circuit of a complete microcomputer system. The only part of this whole system that we haven't seen in the preceding diagrams has been the power switch.

You may at this point be asking, "What can you do with it?" The answer is, not much. With less than 256 words of RAM available in the basic system as shown, you will be very limited. More memory is an absolute necessity. And yet even in this spartan form it can be used. Here is an example.

While working on a UART speed converter, I needed a source of test signals to feed into the UART. Specifically, I needed a constant string of Baudot letters. First, I wired up the simple output circuit of Fig. 10. Using a few of the extra IOT outputs provided by the computer, we control a flip-flop so that a 64X4 instruction sets the flip-flop and a 64X1 resets it. This circuit was built on part of a Radio Shack 44-pin breadboarding card and plugged into one of the three connectors on the back of the computer. Keeping with good habits when using CMOS logic, unused inputs were grounded, and used inputs were connected to Vcc through 10k resistors so they would not be left disconnected if the card is unplugged; this protects them from being zapped by static electricity. Having a simple way of setting and resetting this flip-flop, I next wrote the routine in Program A, which outputs the letter P over and over. The program essentially picks up the bits 01101 for the letter P, adds two stop bits and a start bit, and then starts shifting it through the accumulator, setting or resetting the flip-flop as needed as each following bit is examined. It does this until it runs out of bits, at which time it waits about 87 milliseconds and then starts the whole process again. Since timing loops are used to time out the 22 milliseconds required for each Teletype bit, the very first part of the program turns off the interrupt timer to prevent the control panel interrupts from messing up the timing.

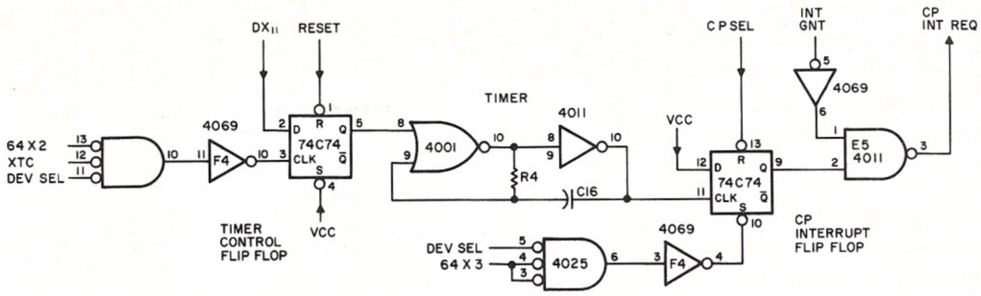


Fig. 8. Interrupt timer and control.

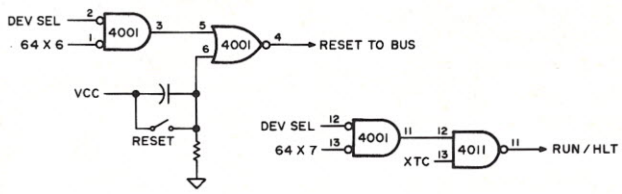


Fig. 9. RUN/HALT and RESET logic.

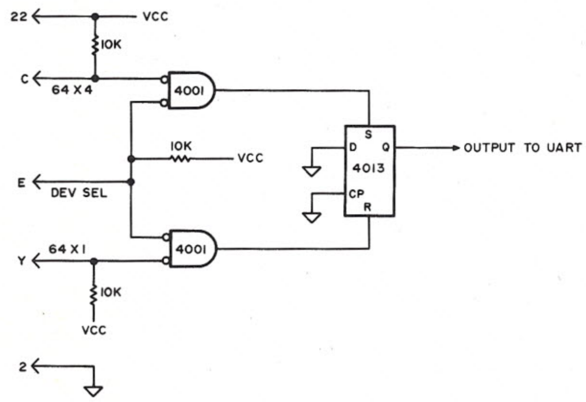


Fig. 10. Simple output interface.

As you can see, this program required 36 (decimal) locations. Hence the memory is definitely large enough to print out an entire test message of perhaps 50 or 100 characters. I suspect that it is also large enough to do a conversion from ASCII to Baudot or vice versa, though I have not tried it.

Conclusions

Both the IM6100 microprocessor and the Intercept Jr. prototyping system/trainer are most interesting devices and quite powerful. There is a potential here for a most useful system if one adds a lot of memory and some input/output devices.

The microprocessor is especially attractive since it can (so they say) run PDP-8e system or application programs. Moreover, Intersil appears to have made arrangements with Digital Equipment Corporation to make the DEC PDP-8e programs available to IM-6100 users. To be exact, there are two types of PDP-8e instructions/applications that the IM6100 cannot duplicate. Its direct memory access (DMA) procedure is a bit different and so the real fast DEC input/output devices would not be usable, though with a few modifications slower DEC input/output devices can be hooked up to the IM6100 and run directly with DEC programs. Second, the PDP-8e has a user flag which is useful for time-sharing; the IM6100 does not. This is a slight pity because DEC has some interesting multiuser BASIC time-sharing systems.

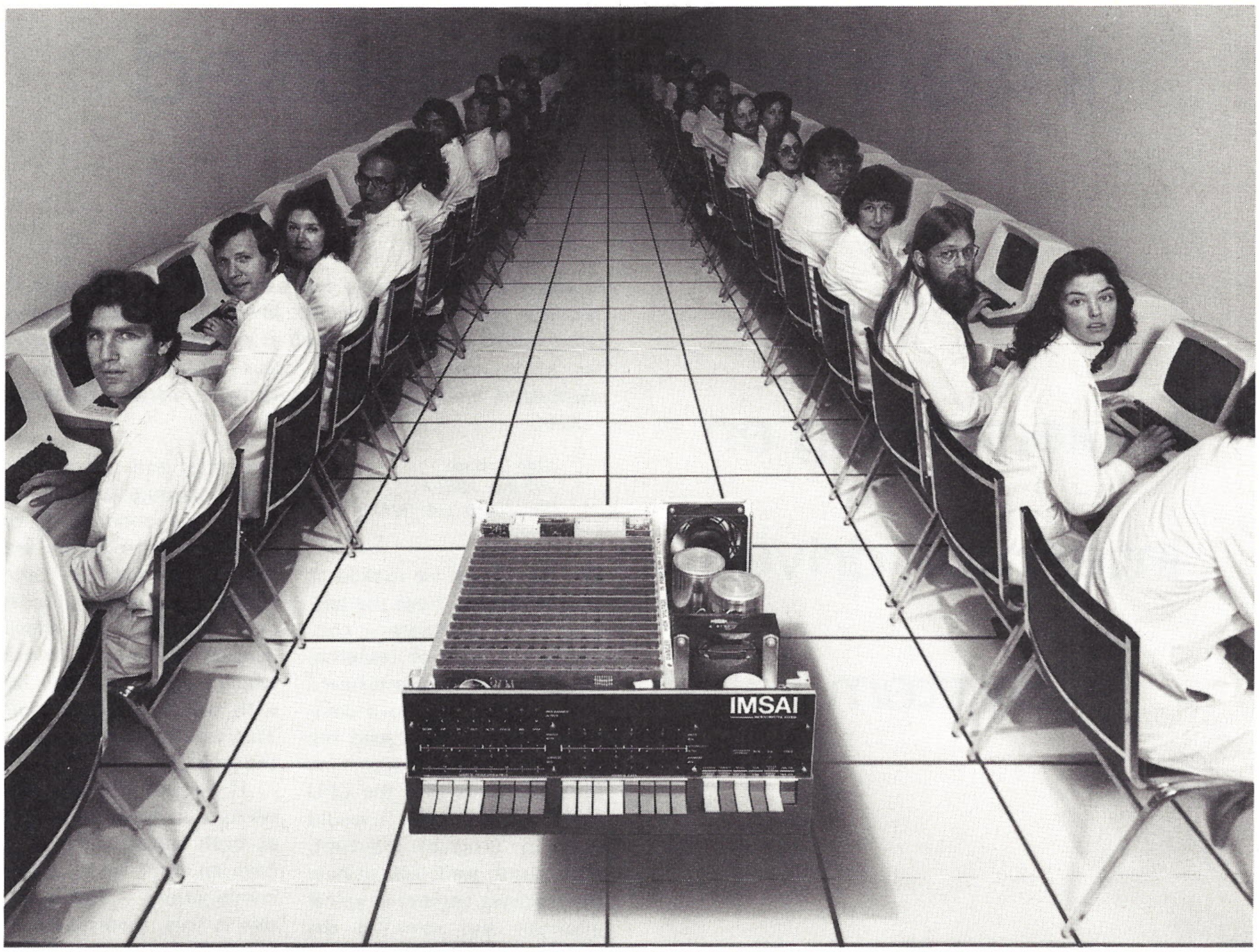
The real limitations exist in the Intercept Jr. system, which puts in some additional constraints. I have already mentioned the problems with the control panel memory and the ROM programming. Another problem is that the optional Teletype interface board requires completely different programming from that commonly used with Teletypes on the PDP-8e. It uses a Peripheral Interface Element (PIE) and cannot be rewired to match the common PDP-8e Teletype instructions. Thus DEC programs using a Teletype or other terminal will not work unless they are changed. In the long run it may be easier to build a different Teletype interface to match the PDP-8e instructions, since the supposed compatibility with the PDP-8e is the Intersil's major (and perhaps only?) advantage over competing microprocessors. ■

```
0200  7200  STRT,  CLA        / Clear accumulator
0201  7001         IAC        / Add 1 (makes DX11 1)
0202  6402         6402       / Turn off interrupt timer
0203  7300  REPT,  CLA CLL    / Clear accumulator and link
0204  1236         TAD LETR   / Get the Baudot letter P
0205  1237         TAD STOP   / Put in two stop bits
0206  0240         AND MASK   / Eliminate extra bits
0207  7004         RAL        / Rotate left (puts in start bit)
0210  3243         DCA TEMP   / Store in TEMP
0211  7300  LOOP,  CLA CLL    / Clear accumulator and link
0212  1243         TAD TEMP   / Get TEMP character
0213  7450         SNA        / Skip next step if not zero
0214  5231         JMP DONE   / Jump if character is all done
0215  7010         RAR        / Rotate next bit into link
0216  3243         DCA TEMP   / Store rest back into TEMP
0217  7430         SZL        / Skip next step if link is 0 (space)
0220  5223         JMP MARK   / Otherwise go to output a mark (1)
0221  6401         6401       / Reset flip-flop for a space (0)
0222  5224         JMP W22    / Go to wait 22 milliseconds
0223  6404  MARK,  6404       / Set flip-flop for a mark (1)
0224  1241  W22    TAD C22    / Get the constant for 22 millisec.
0225  3242         DCA KTR    / . . . and store it in KTR
0226  2242         ISZ KTR    / Add one to KTR and skip if zero
0227  5226         JMP .-1    / Not zero, so go back and repeat
0230  5211         JMP LOOP   / 22 ms over, so do next bit
0231  7200  DONE,  CLA        / Whole character is done, so wait
0232  3242         DCA KTR    / Store 0 into kounter
0233  2242         ISZ KTR    / Add one to KTR and skip if zero
0234  5233         JMP .-1    / Not zero, so go back and repeat
0235  5203         JMP REPT   / Finished waiting; send again

0236  0015  LETR,  0015       / Letter P is binary 01101
0237  0140  STOP,  0140       / Two stop bits (1100000)
0240  0177  MASK,  0177       / 1111111 mask
0241  5764  C22,   5764       / -1036; 1036 reps equal 22 msec.
0242  0000  KTR,   0000       / Kounter
0243  0000  TEMP,  0000       / TEMP storage location
```

Program A. Baudot character output program.

POWER.



IMSAI Introduces the Megabyte Micro.™

The Megabyte Memory

Until today, the largest memory you could fit and address in a single microcomputer CPU was 65K.

Now, IMSAI presents an incredible memory system for micros 16 times more powerful than yesterday's best.

Imagine, a full megabyte of power from sixteen 65K RAM boards.

And, to control all this, the IMSAI Intelligent Memory Manager (IMM), the super control board.

You can write protect blocks throughout the full megabyte. Or, map in 16K blocks.

Plus, preset 16 mapping configurations with protect for high speed transfer or rapid change.

All interrupts are fully vectored, and there's an interrupt if an attempt is made to write into protected memory.

There's even a real "time of day" clock.

65K, 32K and 16K RAM Boards

Until today, the most memory you could plug into a single slot was 16K.

Now, IMSAI presents memory boards in astonishing multiples of sixteen: 65K, 32K and 16K low power, dynamic RAM Boards. They can be used in any S-100 bus computer individually or in combination to form conventional systems up to 65K bytes.

Every board is fast. With "hidden refresh" and *no* "wait state."

The Complete Megabyte Microcomputer System

The IMSAI Megabyte Micro™ is only part of the story. The full system can include dual floppy disks, terminals, plotters, printers and tape cassettes.

IMSAI also offers the finest high level and peripheral software available. Paper tape and Tape Cassette I/O and super Disk Operating Systems. Plus, BASIC and Disk BASIC with more high level languages coming.

Until today, the microcomputer's potential was just something you talked about.

Now, you can put it to work. Powerfully.

GENTLEMEN:
I'm power hungry!
☐ Send 65K RAM Board Kit $2599 ☐ Assembled $3899
☐ Send 32K RAM Board Kit $749 ☐ Assembled $1099
☐ Send 16K RAM Board Kit $449 ☐ Assembled $679
☐ Send IMM ROM Control Kit $299 ☐ Assembled $399
☐ Send IMM EROM Control Kit $499 ☐ Assembled $699
☐ Send full catalog $1.00

Check/MO enclosed. Amt. $ ______
Charge my: ☐ BAC ☐ M/C

\# ______ Exp. Date: ______

Sig. ______

☐ Send name of my nearest IMSAI dealer

Name ______

Company ______ Title ______

Address ______

City ______

State/Zip ______

IMSAI®

IMSAI Manufacturing Corporation
14860 Wicks Blvd.
San Leandro, CA 94577
(415) 483-2093 TWX 910-366-7287

Interrupts Exposed

. . . Part 2:

Implementing an Interrupt-driven System

In this second part of a two-part series on interrupts Dan discusses the software needed to implement some sophisticated real-time processing, such as a home security system. He has some more interesting ideas regarding this type of application, along with some worthwhile software techniques. — John.

Dan LaDage
PO Box 28517
San Diego CA 92128

Last month we looked at interrupts from the perspective of hardware — connecting a set of external sensors to an interface board and attaching this to a computer system. We traced the progress of a signal from one of the sensors into the CPU itself, showing how it would cause a program interrupt. This month we'll look at how the software handles a typical interrupt and some of the problems which arise in interrupt processing.

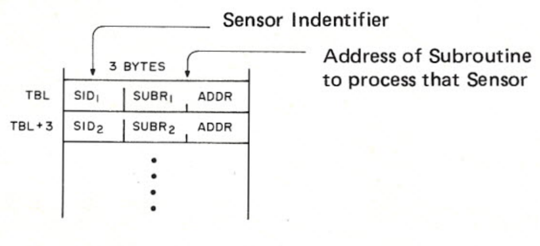
Software has its own organization, and we can view it as being composed of two things: *data structures* (the variables, constants, and tables of data) and *control structures* (the programs that operate upon the data).

Data Structures

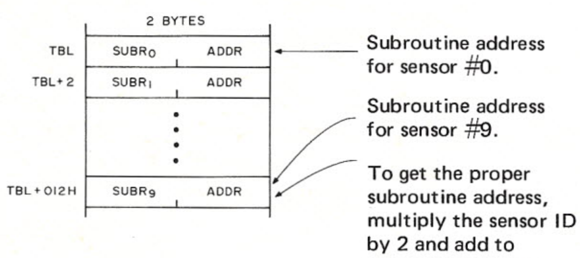
Our purpose is to "identify the sensor and take appropriate action," so a good start would be to set up a table of pairs that associates every sensor-state with the subroutine to handle that condition. Fig. 1a is an example of what such a table would look like. The SID is the state and identifier for some sensor and is a one-byte field. The subroutine location is a two-byte field. We will want to examine the sensor identifier and its state together because we will want to take one action when it goes on and another action when it goes off. Also, we cannot always assume that all sensors are normally at state=0, because some are switches which may be normally open with others normally closed. This would be reflected into the state bit in the SID.

If we have 128 sensors and we have an entry for each one at both states, then we will have an entry for all possible combinations — 256 in all. If this is true, then it would be cheaper (in table space) to just leave the sensor identifier out of the table and use the identifier itself to index into a table of subroutine locations. This method is shown in Fig. 1b. To find the correct subroutine, we would just take the SID and double it and add it to the base location of TBL to get the location of the subroutine address. But this table format is only really useful if we have many devices. If there are only a few devices, say ten or twenty, it would be cheaper to use the format in Fig. 1a.

But there is another way to simplify the table. If you examine the routines in last month's article, you will probably find that there are only a few that are needed, like FIREON, FIREOFF, BURGLARON, BURGLAR-



(a)



(b)

Fig. 1. Possible Data Structures. (a) Paired identifiers and subroutine addresses. (b) Implicit addressing by SID.

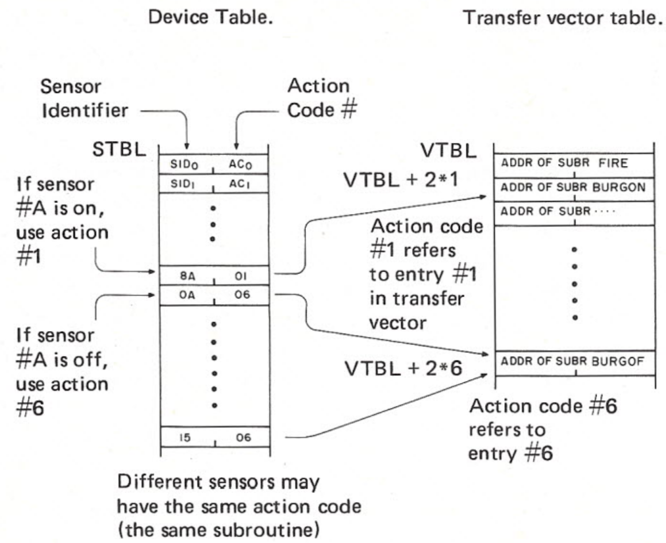


Fig. 2. Cross-referenced Data Tables.

OFF. So if we use either table, the same subroutine addresses will be repeated often. We could replace each subroutine address with a one-byte pseudonym for the subroutine, and this would make the table even smaller. Fig. 2 illustrates this new structure. It has two tables: the device table and the transfer-vector table. The device table is a sequence of entries; each entry is two bytes long and contains, in order, the state and identifier of the sensor (SID) and an action code (AC). The action code identifies the type of action to be taken, so it indirectly names the subroutine to be used for that event. Each action code is associated with one specific subroutine. We can keep the addresses of these subroutines in a table — a transfer vector — and we can use the action code to index into the transfer-vector.

Fig. 2 shows two sample entries. The third entry in the device table shows that if sensor number 0A has state=1, then we should take action number 1, and this is subroutine BURGON. The next entry shows that if sensor 0A has state=0 then we need action type number 6, and routine BURGOF should process it.

These two entries were given as examples because it may be just as important to know when the sensor goes off as to know when it goes on. Computer control of a security system can give it a lot of finesse that it cannot have otherwise. For example, my program might recognize that sensor 0A goes on, and then goes off three seconds later, and that nothing else happens. What probably happened was that the neighbor's dog jumped against the front door and set off the sensor. An ordinary burglar alarm would have been set off, bringing the sheriff and lots of angry neighbors. But my smart computer can recognize such anomalies and make allowances for them. That would take a sophisticated program, of course, but it illustrates one way to build *intelligence* into your system.

Program Operation

Now we have enough basic information — sensors, interface board, and data structures — to build the program itself. Fig. 3 shows a rough flowchart. When the computer gets an interrupt signal from the interface board, it will execute an (RST 1) instruction, which will be a CALL to location 8. At location 8 we will put a direct jump to the ISR routine. ISR will read in the sensor identifier from the I/O port, possibly log the event, scan the device table to find a match with the SID and then CALL the appropriate routine to handle the event. When completed, it will return control to the main program at the point at which it was interrupted. Figs. 2 and 3 in last month's article show the operation of the stack during this sequence. Because of the way this interruption and processing occurs, it can be independent of the main program. So it can take place while you are running your BASIC games, doing your daily accounting, or whatever. If the interrupt processing is short and fast, you won't even notice that it happened — unless, of course, it sets off an alarm!

Logging

One box on the flowchart says "Log the event." This is optional but is a common feature in real-time systems. You may want to keep a permanent record (a log) of all the events that happened; if you have a real-time clock you can also log the time at which they occurred. A simple log would just be a printout on your terminal of the sensor identifier and its state and the time it occurred. You could also print out the type of event (FIRE or BURGLAR) if you wanted to. If you don't have hard-copy capability, then keep the log in a table in memory so you can read it out later. By reading the log you can determine the exact sequence of events that occurred. You can use this to impress the police: "The burglar entered through the kitchen window at 3:43 A.M., then went to the dining room, where he broke into the liquor cabinet; he (or she) stayed there for 51 minutes before going to the living room . . .," and so on. A log is also very useful for testing your sensors and the system. You can trigger them one at a time and then examine the log to verify that they worked correctly.

Caution! Bugs Ahead!

There are several things to watch out for in writing interrupt-service routines. First, we must preserve the integrity of the main program. This means that because the main program does not know that it was interrupted, we must take care to preserve the state of the machine so that it is the same on exit from the ISR as it was when the interrupt happened. To do this we must save (and later restore) all registers and flags; the last thing to do before leaving is to restore them. An ideal place to save them is in the stack. In some microprocessors an interrupt causes all registers to be auto-

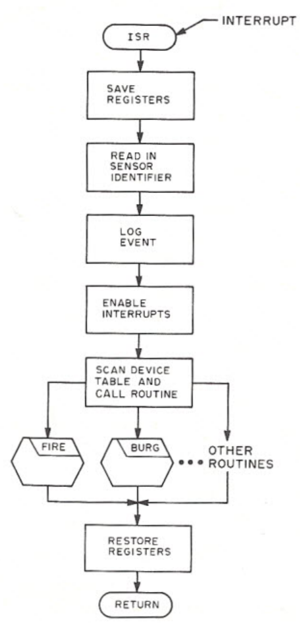


Fig. 3. Program Flow for Interrupt Processing.

matically pushed into the stack, but in 8080 systems we must do this ourselves (in software).

Second, interrupts may occur within interrupts; that is, interrupts may happen while you are still in the process of handling the previous one. The whole interrupt-service processing must be evaluated with this in mind. The safest way is to disable interrupts on entry to ISR (the 8080 does this automatically for you in the RST instruction), and to keep them disabled until you are ready to exit, and then to enable them. This may cause a problem, because if an event occurs while you have interrupts disabled, then the processor will not recognize the signal from the interface board. Depending on how the interface board is designed, you will lose either that new event or the next one. In either case you will lose all events that happen while interrupts are disabled. Because of this we really don't want to disable interrupts for the full term of ISR processing. But it is necessary to disable them while we are doing *critical* code. What is critical code? It is anything that may get mangled if an interrupt occurs, and it is best revealed by the following example.

Suppose we are keeping a log of the events that occur by storing away each sensor identifier and state in a table. They are stored in sequence in the order that the events happen. Let's call the table LOG and let's keep a two-byte variable LOGPTR as a pointer to the next available slot in the LOG table where the event information will be stored. It can be done as in Example 1.

When we log an event, we will place the state and identifier in the next available place in the table and then increment LOGPTR as in Example 2.

Suppose that LOGPTR contains 0400H and we are now logging an event. We execute the instruction at 021B, which loads 0400H into H and L. Right after executing this instruction, an interrupt occurs for the next event. This forces the current processing to be suspended and ISR is reentered and it starts to process the new event. It will eventually come to location 021B and load LOGPTR. But LOGPTR still contains 0400H. So the code will load LOGPTR into H and L, store the new event at location 400, increment the pointer to the value of 0401H and store it back into LOGPTR. Finally the interrupt processing will complete, and it will return to the program that was interrupted. So now we will continue back at location 021E, processing the *previous* event. Because all the registers have been properly saved and restored, H and L will have 0400 in them, and the old event will be in the A register. The program will now proceed to store the old event in location 0400 right on top of the more recent event, and the pointer will then be incremented to 0401 and stored into LOGPTR. We will have actually *lost* the new event by not updating the pointer correctly!

This particular problem can be resolved by disabling interrupts around the critical code as in Example 3.

This will ensure that the variable LOGPTR is properly updated before the next interrupt uses it.

This is only an example of the type of peculiar malfunction that can (and often does) occur in interrupt-service routines. The conditions under which this kind of error can occur are these:

1. The variable (like LOGPTR) survives throughout the processing of multiple events (that is, it is not a temporary variable).
2. Its value changes because of the processing (that is, it is not a read-only constant).
3. Its value cannot be changed by a single instruction. For example, if some variable COUNT were a one-byte variable, we could increment it directly in memory as shown in Example 4. An intervening interrupt would not cause a problem.

But if it takes several instructions to change the variable's value, it is wise to surround the code with Disable/Enable instructions. That is, you should disable interrupts before you load or use the variable, and reenable interrupts after you store the updated value. Even then, there can be peculiar situations involving multiple variables that are very difficult to figure out.

I have spent some amount of space discussing this situation because it is a classical source of errors in interrupt processing. Such bugs can be very difficult to find because they happen rarely and they are nonrepeatable. That is, the error in the logging code will not cause a problem unless an interrupt occurs after instruction 021B and before instruction 0220. This is extremely unlikely but it is possible. What will probably happen is that the system will run fine for months or even years, and then suddenly it will crash. When you try it again, it will run fine, and you can't repeat the error. So you will probably shrug your shoulders and call it a "transient hardware failure." This kind of bug exists today in some very famous operating systems built by major computer manufacturers. They too have trouble finding the bugs, and they are embarrassed when a system crashes. Your system need not be as big and complicated as theirs, however, so your real-time system can be the source of fun as you build it, test it, and watch it run.

And Finally

Interrupt-processing is not painless, but it can be a lot of fun and very rewarding. A real-time system like this security monitor is a miniature version of some of the well-known large-scale systems used for things like spacecraft monitoring and air traffic control. Knowing how these systems work (and how they fail!) is interesting, but the best fun is to build your own. The interface board as described can be used for many things besides just security monitoring, and the software structure is also general purpose. Dream up your own use for the system, and then write and tell the rest of the world about it! ■

```
LOG       DS        100       ;space for logged info
LOGPTR    DW        LOG       ;pointer to next empty slot
```

Example 1.

```
0219   IN      (sensor I/O port)
021B   LHLD    LOGPTR    ;Load address of next spot
021E   MOV     M,A       ;Store event ID in LOG table
021F   INX     H         ;Increment the pointer
0220   SHLD    LOGPTR    ;Store updated pointer
```

Example 2.

```
0218   IN
021A   DI                  ;Disable interrupts
021B   LHLD    LOGPTR
021E   MOV     M,A
021F   INX     H
0220   SHLD    LOGPTR
0223   EI                  ;Enable interrupts
```

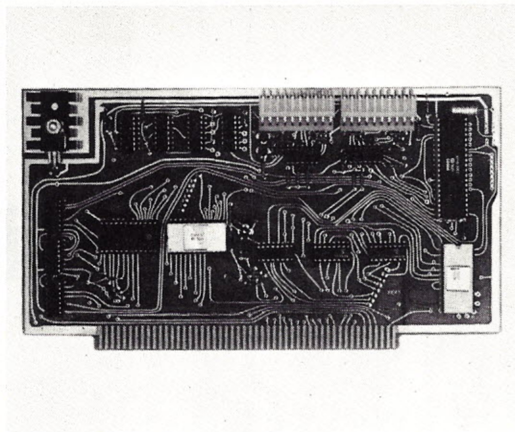
Example 3.

```
LXI    H,COUNT    ;Load address into H and L
INR    M          ;Increment the byte in memory
```

Example 4.

DIGITAL DATA RECORDERS

USING 3M DATA CARTRIDGES



MODEL 3M3 – $199.95

(Price increases to $220 effective 1 April 1977)

2SIO (R) CONTROLLER – $190.00

MODEL 3M3

Featuring the radically new "Uniboard" method of construction for data cartridge drives. The major computer makers are changing to cartridges at a rapid pace because of the freedom from binding and greater data reliability. Operates in the phase encoded self-clocking mode which provides greatly enhanced freedom from speed variation problems and allows 100% tape interchangeability between units.

Uses the 3M Data Cartridge, model DC 300. This cartridge contains 300 feet of .250 tape in a sealed plastic container. Using four tracks you can record nearly 2 megabytes of data on a cartridge.

SPECIFICATIONS:

Full software control of record, play, fast forward and rewind. LED indicates inter-record gaps. EOT and BOT are sensed and automatically shut down recorder. Feedback signals send reset and inter-record gap signals back to the computer so that software searching for inter-record gaps at high speed can be accomplished. Can also be operated manually by means of the switches on top which parallel the software control signals. $199.95 until April 1, 1977. $220.00 after April 1, 1977. Includes Phase Encoder Board (ACI).

FOR 8080, 8085, AND Z80 USERS

Comes complete with software listing for the programs in the 2SIO (R) ROMs. Can be controlled by any of the commonly used I/O boards. Send for complete documentation and interfacing instructions on 3M3 and 2SIO (R) ($3.00). These programs provide full software control.

2SIO (R) CONTROLLER (BOOTSTRAP ELIMINATOR)

This is a complete 8080, 8085, or Z80 system controller. It provides the terminal I/O (RS232, 20 mA, or TTL) and the data cartridge I/O, plus the motor controlling parallel I/O latches. One kilobyte of on board ROM provides turn on and go control of your Altair or Imsai. No more bootstrapping. Loads and Dumps memory in hex on the terminal, formats tape cartridge files, has word processing and paper tape routines. Best of all, it has the search routines to locate files and records by means of six, five, and four letter strings. Just type in the file name and the recorder and software do the rest. Can be used in the BiSync (IBM), BiPhase (Phase Encoded) or NRZ modes with suitable recorders and interfaces. $190, wired and tested; $160, kit form.

AUDIO CASSETTE INTERFACE (ACI)

This is the phase encoding board used in the 3M3. Additional components on the board enable you to use audio recorders in the KC standard or the new PE 2400 (2400 baud) systems. Can also be used for Tarbell if you have an 8251 Intel I/O chip. Required if you use an audio cassette with the 2SIO (R) above. $50, wired and tested; $35, kit form.

For 6800 Users: Software programs and I/O board for SWTP are under development. Limited software available now. Ask for 6800 data with $3.00 Documentation package. These programs will provide full software control.

CARTRIDGE AVAILABILITY

Cartridges are made by 3M, ITC, Wabash and others. They are available at all computer supply houses and most major computer service centers. We can supply them at normal current list prices.

"COMPUTER AID" and "UNIBOARD" are trademarks of the NATIONAL MULTIPLEX CORPORATION. The 3M Data Cartridges are covered by 3M Patents and Marks. "UNIBOARD" Patents Pending.

OVERSEAS: EXPORT VERSION – 220 V – 50 Hz. Write factory or – Megatron, 8011 Putzbrunn, Munchen, Germany; Nippon Automation 5-16-7 Shiba, Minato-Ku, Tokyo; Hobby Data, FACK 20012, Malmo, Sweden; G. Ashbee, 172 Ifield Road, London SW 10-9AG.

For U.P.S. delivery, add $3.00 Overseas and air shipments charges collect. N.J. Residents add 5% Sales Tax. WRITE or CALL for further information. Phone Orders on Master Charge and BankAmericard accepted.

Canadian Distributor:
Trintronics Limited
186 Queen Street West
Toronto, Canada M5V 1Z1
Tel: (416) 598-0262

BANKAMERICARD · master charge

NATIONAL MULTIPLEX CORPORATION

3474 Rand Avenue, South Plainfield NJ 07080, Box 288. Phone (201) 561-3600 TWX 710-997-9530.

Tom Scott
Uncalledfor Productions
700 Bay Rd.
Mill Valley CA 94941

Digital Audio

...Part 2: Generating that Weird Music

I thoroughly enjoyed Tom's article last month on digital audio (as just good general-interest reading) and it occurs to me after reading this month's offering that his whole series will have a lot to offer for those interested in computer-generated music (as well as speech synthesis and a host of other applications). — John.

Last month I careened madly through some basic principles of Analog-to-Digital conversion, with audio signals at the analog end. Together we examined the possibility of a digital recorder which would store a string of numbers that precisely describes the sound waveform rather than store the waveform itself. If you remember, the biggest problem here is the size of the memory required for the storage of even short sounds (more than a *megabyte* per minute of sound).

Sooner or later someone will market a true digital recorder that will store hours of superb fidelity — and it might be sooner than you think. Already one researcher has demonstrated a digitally-encoded tape recorder that promises improvements over the conventional tape recording system. I've heard rumors that one well-known manufacturer of video recorders will market an accessory that would permit their video cassette machine to be used for recording several tracks of very high fidelity sound in a digital format in place of video. As other forms of bulk memory become more realistic in price (that means *cheaper*), digital sound storage by other means than tape could revolutionize the home entertainment industry.

This month's installment will continue our ramble through digital audio. We'll examine some of the hardware that you can buy commercially to get an idea of how far this infant industry has come. As a bonus there's an experiment with a D-to-A card that you can buy for your home computer.

Last month I mentioned some of the uses for a very short time delay in sound studios and concert halls. Digital sound technology has spawned a covey of companies producing black boxes that produce short controllable audio delays, usually less than one second long, for only a few kilobucks per copy. The use of these devices has become so widespread in the last year or two that it's hard to find a pop music record that doesn't feature some sort of sonic effect produced by a digital delay.

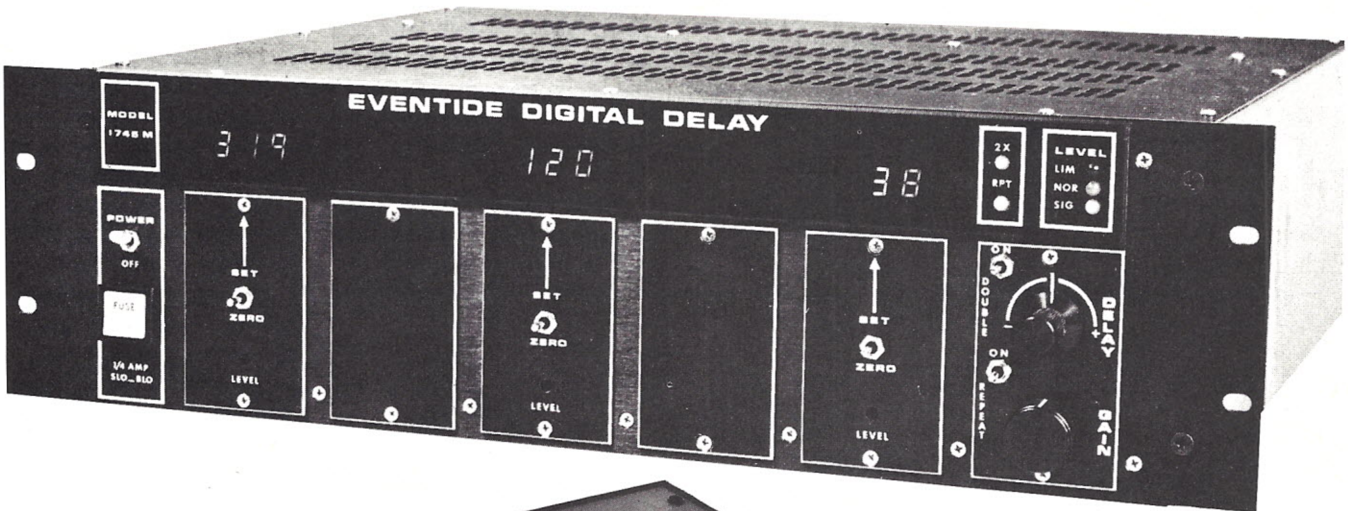
They are, in fact, so popular that *Rolling Stone* (THE pop music newspaper/magazine) ran an article in their February 10, 1977, issue, covering the history and uses of some digital delay (or DDL) devices. [Not being one to duplicate the work of others, I suggest you look up that article, by Ben Sidran, for a very nontechnical, though mostly correct, sprint through the DDL market.] Incidentally, the L of the DDL mnemonic stands for *line*, since the first digital delay replaced long lengths of coaxial cable as a means of achieving a short electrical delay in measuring instruments. Let us examine how the commercial DDLs work.

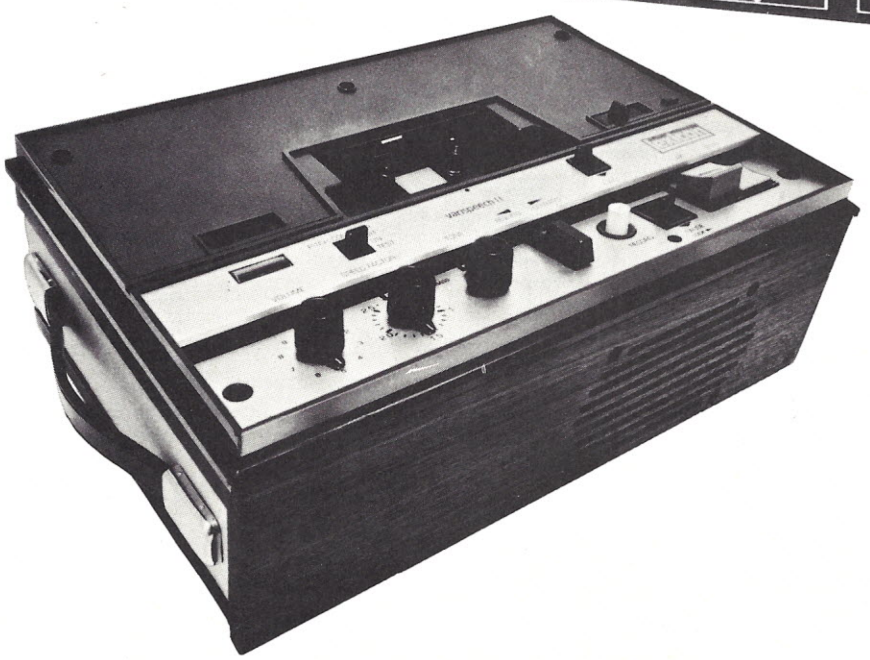
An audio digital delay system starts with an A-to-D converter which takes the input audio and converts it to a stream of binary coded numbers. These numbers describe the instantaneous amplitude of the waveform for a series of points in time (recall last month's article?). Those numbers will be converted back to sound waveforms eventually, but by storing them temporarily, we achieve the time delay we need.

The true digital delay systems I've run across utilize one of two principle circuit

0000	DB	1F		IN, 1F	Input a byte from D+7A port
0002	D3	1F		OUT, 1F	Output the byte to D+7A port
0004	2F			CMA	These two steps output the byte to the
0005	D3	FF		OUT,FF	panel LEDs for visual indication
0007	C3	00	00	JMP,0000	Go back to the beginning

Program A.





Three Digital Audio Devices: 1. The LEXICON Delta-t digital delay is an advanced shift-register-style audio time delay unit. 2. The EVENTIDE CLOCKWORKS 1745M, a RAM memory-style DDL that may be adapted for pitch shifting as well as delay. 3. The Varispeech II, by LEXICON, combines a pitch shifter with a variable speed cassette player to vary the rate of speech while maintaining normal pitch.

elements: shift registers or RAM memory chips.

Shift Register Delays

A shift register is a device that contains storage elements strung together in a long line, 500 or 1000 elements on a single chip. The storage locations work bucket brigade fashion: Information in the form of a 1 or 0 is placed into the first location, and it is then *shifted* to the next location in line as new input data (another 1 or 0) takes its place at the input location. After 500 or so steps down the line, the original information pops out at the far end of this digital pipeline.

The rate of which this step-by-step shifting takes place is set by an oscillator *clock.* Every time the clock pulses, each shift register location dumps its contents into its neighbor location down the line – rather like digital musical chairs. The encoding rate of the A-to-D converter tells us how fast digits arrive at the input of the shift register. This had better be the same as the clocking rate for the shift register or we'll have an information logjam. As a quick review of last month's A-to-D experience, let's consider the rates found in a commercially available DDL.

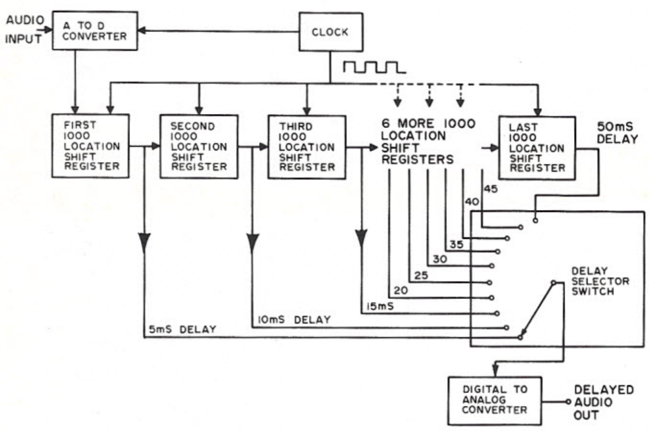


Fig. 1a. A digital delay using shift registers as the storage elements.

The encoding rate for a 10kc audio bandwidth must be at least 20,000 samples of audio each second. The dynamic range of the system is set by the magnitude of the number used to describe the range of possible signal amplitudes. A 1.0 bit number yields a dynamic range of about 70 dB. This means that 200,000 bits of information per second arrive at the input of the shift register. It follows from this that a 1000-bit shift register chip will give us a delay of 1/200th of a second or 5 milliseconds. Ten of these shift register chips in a line will delay a total of 50 milliseconds. If we tap off the output of each chip and allow the D-to-A output stage to switch-select which tap will be the last of the chain, we have created a 5 to 50 millisecond delay adjustable in 5 millisecond steps (see Fig. 1a).

Shift register style delays are still popular, but semiconductor price economics is a wonderful thing. Remember how expensive the first transistor radios were? Even with inflation, transistor AM radios are practically a giveaway item these days. Memory devices are no different. A few years ago semiconductor memory was expensive. Now the price has come down to around a penny a byte, and at least one DDL manufacturer has based his unit on RAM storage with some extraordinary increases in flexibility.

RAM Memory Delays

The Eventide Clockworks model 1745M digital delay system uses forty 2107-type memory chips to hold a total of 16K data bytes of 10 bits each. Audio is sampled and digitized at a rate of 50,000 samples per second. A little arithmetic will tell you that their 16K memory provides 320 milliseconds (1/3 of a second) worth of sound. Eventide hasn't used a microprocessor chip, preferring to control the memory storage and retrieval with a processor made up of CMOS and TTL chips. But their processor performs the same functions that a microprocessor might.

A register is used to store the memory address where the last byte of data was stored. Since each memory location represents 1/50,000 of a second, the *processor* can calculate just how far back in the memory we need to go for a specified time delay. Let's specify 1/100 of a second. Again a bit of arithmetic. If we subtract 500 memory locations from the address register, we arrive at the memory location which, when read out and converted back to sound, will give us back the sound that was input 1/100 of a second ago (plus a little time for decoding). See Fig. 1b.

Can you visualize that? The input stage is constantly sweeping through the memory, depositing new data while the output decoder is trailing along, a specific delay distance behind, pulling data out and reconverting it to

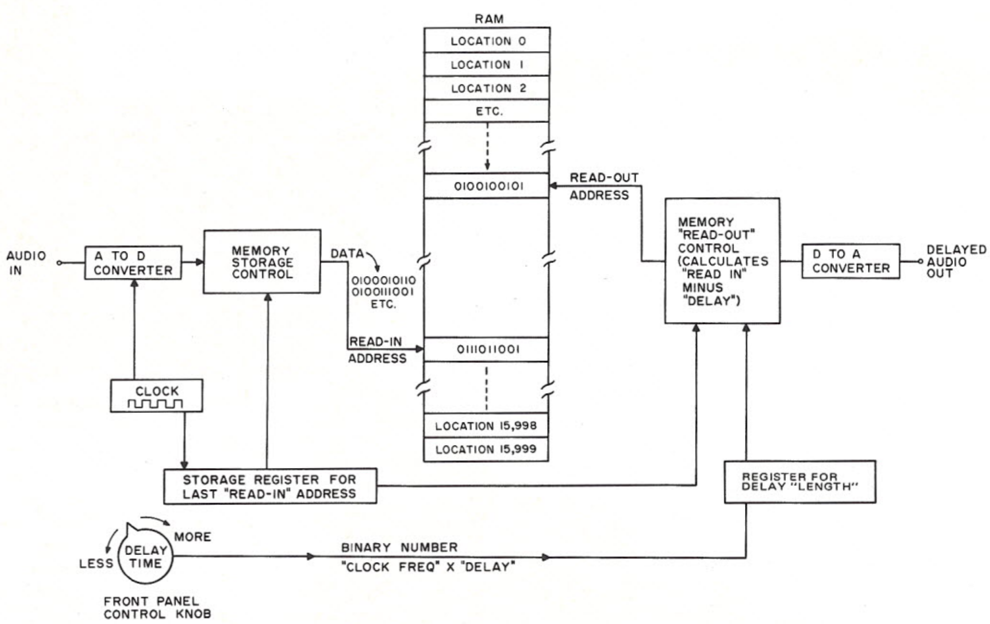


Fig. 1b. A digital delay using RAM storage elements. Note that the readout control calculates *the readout address by subtracting the delay from the read-in address.*

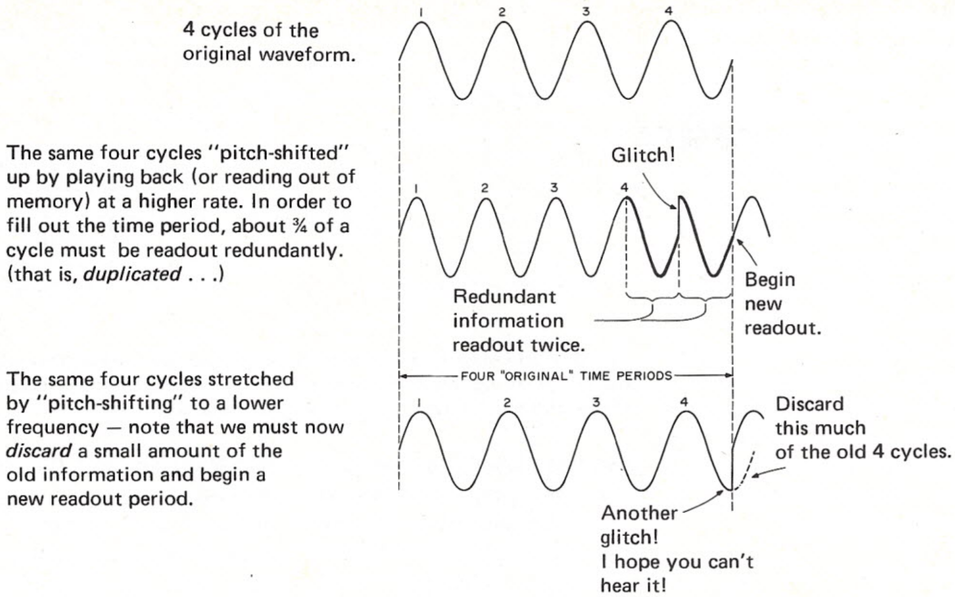


Fig. 2. If pitch *is shifted, information must be either duplicated or discarded to keep the* rate, *or the time period, the same.*

sound. Of course, the delay distance must stay constant so the output doesn't catch up to the input, right? "The results could be catastrophic," you say? Sure, we'd like to keep from overfilling or overdrawing our memory bank. But, just for a second, stop and consider what it would mean to us to be able to run data into the memory at one rate and out at another rate. It would mean we could change the pitch and rate of sound at will — just like changing the playback speed of a tape recorder or a record player.

Pitch Change

If I take a tape recording made at 7-1/2 inches per second and play it back to you at 8 inches per second, what do you observe? The pitch of the reproduced music will be higher and the tempo will be quicker. Remember the Chipmunks? For those novelty records someone sang in a normal voice at 1/2 the tempo of the finished tune. The tape was then played at double speed. The pitch jumped up an octave, producing cute chipmunk voices. [Doubling the frequency of a tone is equal to a one octave musical jump.] The tempo doubled from half-speed to the beat specified by the tune's arranger. This may seem like a frivolous example, but it illustrates clearly that the pitch and the tempo of recorded material are intimately related and not easily separated.

There are, however, times when it would be convenient to change the pitch of music *without* changing the tempo: like lowering the pitch of a musical track so that it falls into a certain singer's range without turning the song into a dirge. Or how about spoken-word recordings or talking books for the blind? It is sometimes expedient to speed the rate of delivery far beyond the capability of most announcers and yet not change the pitch of the speaking voice.

During the 1960s experiments were made with special tape recorders with rotating playback heads which would leave out tiny sections of the program material to speed the rate or tempo without changing the pitch. The same machines could also play back at a higher pitch and an unchanged rate by repeating and overlapping short segments of the program material. These devices did work, but they were very mechanical and delicate. Furthermore, they introduced a lot of noise and spurious modulations into the playback sound. Now that digital audio is here, independent pitch and rate changing can be done more effectively and easily. A fairly simple variation on the basic RAM type of DDL can allow us to change pitch while holding the rate constant.

Let's imagine that a musical passage is being digitally encoded and stored at a *fixed* rate into the memory of a delay device. To raise the pitch we must let most of the memory fill up to form a buffer zone. Then we begin reading out and decoding at a *faster* rate (just like playing the tape faster). But you can easily envision that the *readout* location must eventually catch up to the *read-in* location. To avoid this, small groups of memory locations are read out twice. These short bursts of redundant information will be too short to notice in the output audio, but if we choose our numbers correctly, the readout location will always trail the read-in. What we've done is *fill out* the decoded output with repeated information so that the output will seem to have the same tempo or rate as the original input. The frequency or pitch of the output, however, will be perceived as higher than the original.

To clarify this process, let's make an example of the related case of pitch *lowering.* Say our audio is an organ note, middle A, with a frequency of 440 Hertz (that's cycles per second for the old school). We sample and encode at a rate of 20,000 samples per second, storing 20 bytes into the memory every millisecond. If we read out at the lower rate of 18 bytes per millisecond, we will output a tone with the frequency 10% lower than 440, or 396 Hz. But since we are putting more into the memory than we are taking out, our memory overfloweth unless we get rid of some chunks of memory input. To prevent overflow we must discard some data every few milliseconds by emptying the memory and starting to fill it again.

Fig. 2 shows how redundant information is added to the output when pitch is shifted higher, or removed when pitch is lowered. That same figure reveals a problem though. What happens if the inserted information doesn't fit? Note that a step or discontinuity can occur in the output waveform at the point where information has been inserted or deleted. Does this glitch matter? This depends on the application and the program material. Glitches in a classical violin solo would be disastrous. Spoken word or percussive music might be none the worse for a few steps or clicks.

For applications where glitches are a problem, there are techniques that can render them inaudible. We might have the control hardware look for gaps in the program material and add or delete pauses there. Hardware or software could be designed to remove or insert only sections of waveform that begin and end on *zero crossings* (zero voltage points). Or, we might design a smoothing filter to remove the higher frequency components of the glitch, making it less obtrusive.

Changing only the *rate* of recorded sound requires one more step, since the program material will now occupy a longer or shorter *time* span. To increase the tempo of a piece of music, a tape of the piece is played at increased speed — the pitch and the rate *both* go up. Next, a digital pitch shift is performed on the sped-up tape and presto! — the pitch is lowered back to normal but the rate remains at the faster tempo. Lexicon, a manufacturer of digital delay units, sells a combination pitch shifter and variable speed cassette deck that simultaneously performs the above process. They utilize a proprietary Intelligent Digital Splicing so that pitch shifting glitches are not audible. Their unit is ideal for spoken word recordings: Audition of recorded verbal material may be sped up to well above the fastest reading rate and yet the pitch of the reader's voice stays normal.

By this time I hope you have concluded as I did, that microcomputer control of audio delays and pitch changing is within the reach of the audio computophile who likes the challenge of a bit of experimentation. So, on with experimentation.

Another A-to-D-to-A Device!

Last month I pointed out a chip that might form the basis for a hobbyist's Analog to Digital Conversion scheme. If that route is not easy enough for you, how about this one? A neighbor of mine, Jim Coe, is the sound technician for the Jefferson Starship rock group. He's also an avid audio computophile. He showed me that there is a ready-to-use, Altair bus compatible card available to get you started right away in digitized audio.

It's the Cromemco D+7A™ input/output card, and though not designed specifically for sound, it will linearly encode and decode 8-bit digital audio! The reproduction is about as good as that of an AM transistor radio, and it requires no modifications at all for your first experiment. Simply connect a source of audio, say your old faithful cassette player, to analog input number 7 (there are 7 available) and connect analog output number 7 to one input of your stero system. If you've got an oscilloscope, hook that across the output, too. It will be very instructive to see, as well as hear, what's coming out of the D-to-A section.

Turn It On

With the card in your Imsai or Altair and the power on, Program A will read in one data word from the output of the A-to-D section into the accumulator. Then it will output the same word to the D-to-A section and jump back to the beginning again. No delay except for a few computer cycles, no pitch change or other fancy stuff, but you're on your way. You've got input and output.

Right away you'll be able to think of improvements. There will be quite a bit of background noise on the output when small signals are applied. That's to be expected, since the 8-bit linear encoding can't really do justice to tiny signals. Even the playback hiss from the cassette player will drive the least significant bits of the A-to-D crazy. If you have a Dolby or DBX on your tape deck, you might be able to noise-reduce the A-to-D-to-A chain. If your audio source is too good, you may hear some dissonant modulation products mixed in with the output audio. That is, other tones showing up in the output that weren't present in the input.

A glance at the D+7A™ schematic and a quick rereading of last month's article will provide the tipoff. There is no input filter to keep out high frequencies that are above 1/2 the sampling rate of the A-to-D. If your

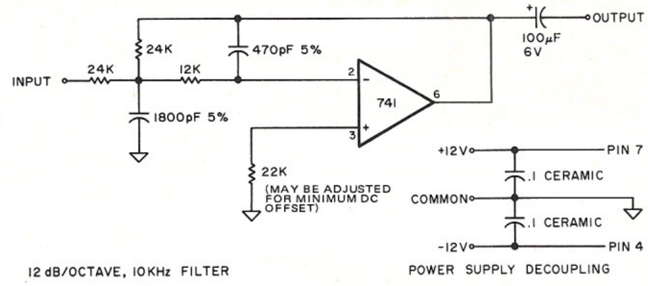


Fig. 3. A low pass filter suitable for both input and output filtering for the D+7A™ I/O card. Pin numbers are specified for the 8-pin package. (From IC OP AMP Cookbook, *Walter G. Jung, Sams, 1974)*

cassette player is a cheapie, it may not play back enough high frequencies to do any damage, but an input filter is a good safeguard. Also, there's no output filter to smooth off the D-to-A reconversion. The oscilloscope reveals that the output is made up of little steps instead of a smooth continuous line. Fig. 3 shows a filter that can be made with cheap op-amps that should provide a cure for both problems, though as of this writing, it hasn't been tried out.

If any readers have experimented with the D+7A™ or have discovered other A-to-D systems that can work with home computers, let us know what you've found out. What sorts of programs have you written? What end uses have you envisioned for your system? Having touched ground there briefly with a little hardware, let's get back to the blue sky and I'll tell you about a commercially available, solid state recorder about the size of a paperback book!

A Solid State Recorder

The module is called a Voice Storage Unit, and it is produced by Comex Systems of Hudson, New Hampshire. From their advertising literature, it seems to be a true digital recorder with no tapes or moving parts. The unit will store up to 12.5 seconds of sound at a price of about $100 per second of storage. They claim "Excellent Vocal Quality" but cite no specifications of frequency response or dynamic range. Let's guess 300-3000 Hz and 65 dB of dynamic range. Comex doesn't say exactly how much RAM memory they use but, now that you know a little about sampling theory and A-to-D conversion, I'll bet you could form a pretty good estimate.

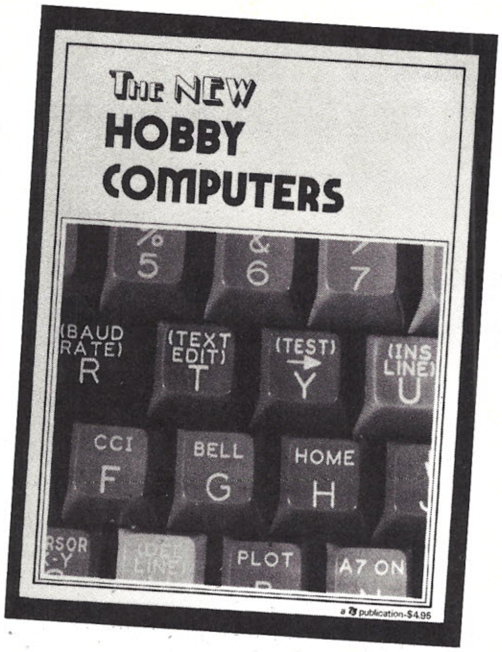
The same company also sells a variation on that theme which they call a Random Access Voice Storage Unit, with 12-second capacity. It stores 16 short (2/3 of a second) messages like: "Hello," "seventy," "three," "emergency," "thank you". Whatever you record. It will then output any combination of its vocabulary as you apply four-bit binary address codes to its control terminals. I wonder if you can make your 8080 do that? The usual applications are repeated airport announcements or phone messages. Since there is no tape to wear out, the thing can talk forever. Hobbyists of course will immediately flash on talking home robots, barking dog doorbells, or a computer terminal that wakes you every morning with "Time to get up, Master" . . . but you get the idea.

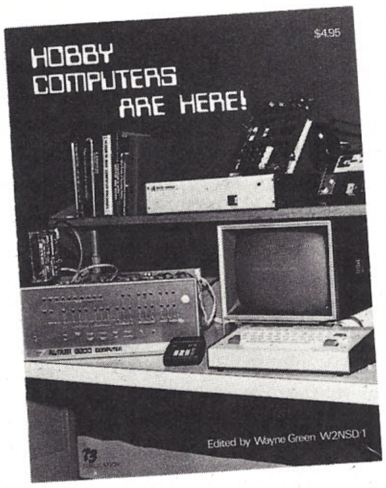
Next Month

Next month I hope to tell you about Digital Signal Processing. It's wonderful to be able to trap and store sounds digitally, but the best part is yet to come. Once the sound is stored as numbers and before we send them back out as sound, it is possible to do things digitally that audio engineers a generation ago never dreamed of. ■

The NEW HOBBY COMPUTERS

This book takes it from where "Hobby Computers Are Here" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards, a Baudot monitor/editor system, an audible logic probe for finding those tough problems, a ham's computer, a computer QSO machine . . . and much, much more! Everything of interest is there in one volume, ready to be enjoyed by you. Don't miss this tremendous value! Only $4.95





"It's the first book I've ever read about computers that I can understand . . ."

HOBBY COMPUTERS ARE HERE helps the beginner get into the world of microcomputers. Some chapters: What's a Computer?, Is Digital New?, How Computers Figure, Computer Languages, How Gates Work, TTL – Best Logic Yet, Ins and Outs of TTL, Flip-Flops Exposed, New Cassette System Standard, Build this TVT, Using Surplus Keyboards, Morse to RTTY Converter, ASCII to Baudot via a PROM, PLUS reprints of some of the 73 editorials on computers. Don't miss out any longer on the fun of hobby computing and the fantastic applications of these incredible devices! $4.95

Mail in coupon or phone TOLL FREE (800) 258-5473

Please send me: ____ The New Hobby Computers @ $4.95

____ Hobby Computers Are Here! @ $4.95

Total order $ ________

$____ enclosed. ☐ Cash ☐ Check ☐ Money order

Bill: ☐ Master Charge ☐ BankAmericard ☐ American Express

Card # ____________________ Interbank # ________

Expiration date ______ Signature ____________________

Name ____________________

Address ____________________

City ____________________ State ______ Zip ______

KILOBAUD • DEPT. HC • PETERBOROUGH NH 03458 KB/5/77

Stephen Pereira, President
Microcomputers, Inc.
539 Amherst St.
Nashua NH 03060

Now It's Imsai BASIC!

Along with some good introductory material on the BASIC language, Stephen has presented us with a good description of Imsai's BASIC in the following article. I would suspect there are a number of people who would like to see a comprehensive and objective evaluation and comparison of this BASIC along with MITS' and Southwest Technical Product's version. Any volunteers? Should be an interesting project. – John.

The BASIC language is a conversational programming language which uses simple English-type statements and familiar mathematical notation to perform operations. BASIC is one of the simplest computer languages to learn, and once learned, provides advanced techniques to perform intricate data manipulations, and efficient problem expression.

The 8K subset of DEC BASIC PLUS that runs on the Imsai 8080 microcomputer system is an interpretive program. That is, each line of a users program is interpreted into machine language by the executive program each time it is executed.

A BASIC program is composed of lines of statements containing instructions to the BASIC interpreter. Each line of the program begins with a line number that identifies that line as a statement and indicates the order of statement execution relative to other lines in the program. Each statement starts with an English word specifying the type of operation to be performed.

All BASIC statements and computations must be written on a single line. Statements cannot be continued on a following line. More than one statement, however, can be written on a single line when each statement after the first is preceded by a back slash (\ ). Line numbers may range from 0 to 9999.

Constants and Variables

BASIC treats all numbers (real and integer) as decimal numbers. The advantage to this is that any number or symbol can be used in any mathematical expression without regard to its type. Numbers must be within the range 10^{-18} to 10^{+18}.

LET – Assigns the value of an expression to the specified variable.
LET A=SQR (3)
DIM – Reserves space in memory for arrays according to the subscripts specified. This version of BASIC will allow a maximum of 255 subscripts per variable.
DIM A (100)
DIM X (50, 50)
DATA – Used in conjunction with the READ statement to input listed data into an executing program.
DATA 1, 2, 3, 4, 5
READ – Assigns values listed in DATA statements to the specified variables.
READ A, B, C
RESTORE – Resets the data pointer to the beginning of the data file.
PRINT – Prints the values of the specified expressions on the terminal.
PRINT A, B
INPUT – Queries the terminal for data to assign to a numeric or alphabetic variable.
INPUT A
FOR . . . TO – Sets up a loop to be executed the specified number of times. Unless specified, the step is 1.
FOR I=1 to 10
FOR J=0 to 100 STEP 2
NEXT – Placed at the end of a FOR loop to return control to the FOR statement.
NEXT I
IF . . . THEN – Conditionally executes the specified statement or transfers control to the specified line number. When the condition is not satisfied the control is passed to the next sequential line number. The expressions and relational operators must be all numeric or all string.
IF X < 3 THEN 100
IF A=10 THEN PRINT "FINISHED"
GOTO – Unconditionally transfers control to the specified line number.
ON . . . GOTO – Conditionally transfers control to one of the line numbers in the specified list. The value of the expression determines which line number control is passed to.
ON X GOTO 30, 40, 50
(if x=1 GOTO line 30, if x=2 GOTO line 40, if x=3 GOTO line 50)
GOSUB – Unconditionally transfers control to a specified line of subroutine.
GOSUB 3000
RETURN – Terminates a subroutine and returns control to the statement following the last executed GOSUB statement.
ON . . . GOSUB – Conditionally transfers control to the subroutine at one line number specified in the list. The value of the expression determines the line number to which control is transferred.
ON A GOSUB 100, 200, 300
(If A=1 GOTO SUBROUTINE @ line 100, if A=2 GOTO SUBROUTINE @ 200, if A=3 GOTO SUBROUTINE @ LINE 300).
STOP – Terminates execution of program. Placed at the logical end of the program.
END – Statement placed at the physical end of the program to show the end of the listing (has no other value than a remark).
REM – Indicates a remark line and will not be executed as part of the program. Will be printed as part of the listing.
DEF FN – Defines a user function
DEF FNA (X) = SQR (X)
∴ FNA (4) = 2
CHANGE – Transforms a character string to a list of numeric values to character string.
CHANGE A$ TO X

Table 1. Statements

BASIC uses the most efficient format for printing a number, according to its size. It automatically suppresses leading and trailing zeros in integer and decimal numbers and formats all exponential numbers. Floating point format is used when storing and calculating numbers.

BASIC also processes information in the form of strings. A string is a sequence of alphabetic or numeric characters treated as unit. A string constant is a list of characters enclosed in quotes.

This 8K BASIC interpreter recognizes three types of variables: numeric, subscripted numeric, and string. A numeric variable is an algebraic symbol representing a number and is formed by a single letter or a single letter optionally followed by a single digit. For example, I, B3, or X.

Subscripted variables provide additional computing capability for dealing with lists, tables, matrices, or any set of related variables. Variables are allowed with one or two subscripts. For example, a list might be described as A(I) where it goes from 0 to 5: A(0), A(1), A(2), A(3), A(4), A(5). This allows reference to each of the six elements in the list,and can be considered a one-dimensional algebraic matrix. Two-dimensional matrices are also allowed, i.e., x(6,3).

Any variable name followed by a dollar sign ($) character indicates a string variable. For example A$, C7$.

The user can assign values to variables by indicating the values in a LET statement, by entering the value as data in an INPUT statement, or by a READ statement. The value of the variable does not change until the next time a statement is encountered that contains a new value for that variable.

Operators

BASIC performs addition, subtraction, multiplication, division, and exponentiation. The five operators used in writing the most familiar formulas are:

+	A+B	Add B to A
-	A-B	Subtract B from A
*	A*B	Multiply A by B
/	A/B	Divide A by B
↑	A↑B	Raise A to the Bth power

Relational operators allow comparison of two values and are used to compare arithmetic expressions or strings in an If-Then statement. The relational operators are:

= Equals (alphabetically or numerically equal)

< Less than (alphabetically or numerically precedes)

<= Less than or equal (precedes or equal)

> Greater than (alphabetically or numerically follows)

>= Greater than or equal (equals or follows)

< > Not equal (not alphabetically or numerically equal)

Statements

The summary of Imsai 8K BASIC statements gives an explanation of each statement's use (see Tables 1-4).

Immediate Mode of Execution

It is not necessary to write a complete program to use Imsai 8K BASIC. Almost all BASIC statements can be executed in immediate mode. This facility makes BASIC an extremely powerful calculator.

The BASIC interpreter distinguishes between those lines entered for immediate execution and those entered for later execution by the absence or presence of a line number. Statements which begin with a line number are stored. Those without line numbers are executed immediately.

The immediate mode operation may be useful for program debugging and for desk calculation problems.

To facilitate program debugging, STOP statements can be placed throughout the program. When the program is run, each STOP statement will cause the program to halt. Data values can then be examined and modified in the immediate mode. The immediate mode statement CONT is used to continue program execution.

If a STOP statement occurs in the middle of a FOR . . . TO loop, modifications should not be made to the section of program preceding the FOR statement. If CTRL/C is used to halt program execution, the GOTO command can be used to continue execution at the line where the execution was halted.

When using immediate mode, nearly all the standard statements can be used to generate or print immediate mode results. Multiple statements per line can also be used in the immediate mode.

Program loops are therefore possible in multiple statement line. Thus a table of square roots can be produced as in Table 5.

Certain statements, while not illegal, make no sense when used in immediate mode, such as DEF, DIM, DATA, IF . . . THEN. Also, function references in immediate mode are illegal unless the program containing the definition was previously executed. ■

ABS — Returns the absolute value of the expression
ATN — Returns the arctangent of an angle in radians
COS — Returns the cosine of an expression in radians
COT — Returns the cotangent of an expression in radians
EXP — Returns the value of the constant e(2.71828) raised to the power of the following expression
LOG — Returns the natural logarithm of an expression
INT — Returns the whole number value of the expression ∴ INT (PI) = 3
PI — Returns the value of PI (3.14159)
RND — Returns a random number between 0 & 1
SGN — Returns a value indicating the sign of an expression in radians
SIN — Returns the sine of an expression in radians
SQR — Returns the square root of an expression
TAN — Returns the tangent of an expression in radians

Table 2. Arithmetic Functions

ASC — Returns the ASCII code in decimal of a 1-character string
CHR — Generates a 1-character string from the lower order 8 bits of the integer value of the expression
LEN — Returns the number of characters in a given string
POS — Searches for and returns the position of the first occurrence of a substring in a string
STR — Returns the string which represents the numeric value of the given expression
VAL — Returns the value of the decimal number contained in the given string expression
LEFT — Returns the left-most characters of a string determined by a set of parameters
RIGHT — Returns the right-most characters of a string determined by a set of parameters
MID — Returns the middle characters of a string determined by a set of parameters
NUM — Returns a string of numeric characters as it would be output by a PRINT statement

Table 3. String Functions

TAPE — Tells executive to expect a program input from paper tape
FREE — Returns the amount in bytes of free space left in memory for user program storage
OUT — Outputs 8 bits to any assigned port out PORTLOC, VARIABLE
INP — By use of LET statement LET X = INP (PORTLOC) inputs 8 bits from any port
RUBOUT — Deletes the last character typed
CTRL/C — Terminates program execution at end of current line
CONT — Continue from STOP statement
NEW — Clears the user area in memory for a new program
LIST — Types to the terminal the current user program
RUN — Executes the current user program
SAVE — Types to paper tape the current user program with leader and trailer

Table 4. Commands

```
:FOR I = 1 TO 10\PRINT I, SQR(I)\NEXT I
```

1	1	6	2.44949
2	1.41421	7	2.64575
3	1.73205	8	2.82843
4	2	9	3
5	2.23607	10	3.16228

Table 5.

David E. Stanfield
3408 Catalina Dr.
Atlanta GA 30341

Bridging the Gap

...tips on turning an application into a program

As Dave gets into a discussion of the differences between professional and amateur computer types it becomes apparent that he's speaking as an insider of both groups. Some of the points he brings out are quite interesting (and thought-provoking). The other two sections of his article deal with an introduction to programming and programming languages, followed by a very good example of analyzing a problem, defining the solution, and generating a flowchart prior to coding (using a tic-tac-toe game). — John.

Computers are here to stay. Large ones. Small ones. Slow and fast. Every year will see their numbers increase. A very interesting aspect of this computer explosion is the fact that an evergrowing number of ordinary people are taking a computer home and using it for their own pleasure.

While many have already made the plunge into personal computing, a much greater number are holding back. Certainly cost is a factor in this hesitation but the fact is that a decent system can be had for about the cost of a color console and some of the streamlined models cost no more than a black and white set.

Amateur vs. professional programmers

Though cost is an important consideration, I believe that fear of programming plays a much larger role in keeping people from buying their own computer. To the average person who has never even seen a computer, the whole idea sounds far too complex. After all, we've all heard stories of some computer sending out a $13,695 utility bill to some unsuspecting customer. Since such computers are controlled by programmers sporting degrees and diplomas, how can we hope to compete?

Obviously, we can't match the efforts of these super programmers. What may not be as obvious is that it doesn't matter. Professional programmers are paid salaries that range from average to very good. But the point is that they must perform. They must meet deadlines, work on programs they don't care about and try to keep a lot of different people happy.

The home programmer doesn't have to worry about any of these things. If he wants, he can drop a program for a month while he works on his lawn. If he weren't interested in the program, he wouldn't be working on it in the first place. And in most cases, the program will be strictly for his own use so there won't be someone looking over his shoulder to see if he's finished.

Additionally, most professional programmers are so pushed for time that the best they can do is try to meet the program objectives within the limitations of the computer system. They are too busy to write alternate programs that are more efficient, operate faster or, use less memory. The most they can usually manage is to patch an existing program to get around some problem that bothers the customer.

In these respects, the amateur has all the advantages. His programs are not so critical that if he takes a couple of days to polish them fifty people will be thrown out of work. And most of his programs will be short enough that even major program revisions will not consume vast amounts of time.

Another significant advantage possessed by the home programmer lies in his ownership of the finished program. In most cases the professional's programs belong to his employer. These programs may be highly original, capable of saving the user large amounts of money and in every respect a very valuable property.

For the programmer this translates into another paycheck or two and an occasional word of thanks. His employer may use these programs and reap the savings they generate for many years while the programmer is forced to continously produce new programs just to stay employed.

The home programmer is in an altogether different situation. Perhaps it may sound pretentious, but the truth is that good programming is highly creative and finished programs are treated much the same as are other forms of creativity. Generally speaking, the creator of a program owns it and can sell, license, or give it away.

It is unlikely that your programming efforts will make you rich but there is nothing to stop you from trying. Already several magazines pay for original programs that will interest their readers and a number of people are selling copies of their programs thru classified ads. Beyond these areas you are free to create your own markets. The entire home computer market is still in its infancy and there is a lot of room for those who offer potential customers goods and services at reasonable prices. As is true of all markets, a unique and exciting product will be able to command higher prices.

I hope that by now I've convinced you that in many areas, the home programmer has some very real advantages over his professional counterpart. At the same time, I'm aware that I've so far sidestepped the obvious advantages of the professional programmer. His combination of in-depth training and accumulated experience simply cannot be overlooked.

While these advantages seem impossibly imposing, they assume quite reasonable proportions when examined in the proper perspective. If we compare the complexity of the computer systems and the tasks to be programmed, we find that the professional needs all of his training and experience while the amateur can do very well with more modest skills.

The professional programs systems consisting of tape drives, disk drives, their controllers, printers, typewriters, external memory and a central processor. Typical systems have ten or so machines and some systems have literally dozens. He will be writing programs involving thousands of accounts, hundreds of employees, constantly changing inventories and things like taxes, discounts and delivery schedules.

In comparison, the amateur at home will probably have a processor with memory, a TV typewriter and maybe a cassette recorder with a controller. With this system he will be able to write programs that will keep up with his checking account or play games of moderate difficulty. In other words, the professional programmer has to use every bit of his training and acquired skills in order to get his programs to function correctly. Most systems used by amateurs and the tasks they will be programming will be several orders of magnitude simpler than those of the professional.

What this means is that the amateur can function as an effective programmer for his home system with far less training and experience than is needed by the professional. Certainly he will encounter difficulties but so do the pros.

Programming and levels of languages

Having shown some of the advantages of the home programmer, I'd like to explain a little about what programming is, explore the area of program languages, and thru the use of an example, show some of the thought processes involved in programming. I won't be teaching you to program anything, but I will be trying to provide some useful background information.

Simply put, programming is instructing a computer to do a given job. Computers are very dumb and incredibly fast. It is the latter feature which makes them so valuable but it is the first that makes programming necessary.

Therefore, what the programmer does is to thoroughly define the task to be performed and then break it up into a series of discreet operations the computer can handle one at a time. Computers can be programmed to ask questions such as, "Is the account balance greater than zero?" Depending upon the answer, it can choose an appropriate course of action. Such decisions must be carefully laid out in the program, but this ability to respond to different situations gives the computer tremendous flexibility.

The concept of program languages is a major hurdle for many beginning programmers. They hear a lot of people use terms like BASIC, COBOL, FORTRAN and Machine Language. In many cases, a few minutes listening to these experts will convince almost anyone that learning one of these magic languages will turn them into a programmer. As there is some truth and some fiction involved here, we should explore the area for ourselves.

Our starting point is the fact that the computer itself can only understand electrical impulses. If you were to verbally command the computer to add three and four, nothing would happen. It isn't built to *hear* sound. Instead, we are forced to use properly coded electrical impulses when we wish to communicate with computers.

Without too much difficulty, we can set up some indicators to give a visual readout of the combinations of impluses being processed by the computer. Several computers use lamps or light emitting diodes to accomplish this. While useful, this sort of system is very hard for most people to decipher.

What we are discussing here is pure Machine Language. With a few simple switches and lamps, we can use Machine Language to get information into and out of a computer. When people are truly programming in Machine Language, they are actually using the only language the computer understands. However, this system is tedious, slow and hard to use.

Generally speaking, higher level programming languages are programs set up within the computer which allow it to take agreed-upon commands and transform them into the Machine Language codes it really understands. At a simple level, the programmer can use JMP instead of 104 to cause an unconditional Jump when programming an 8008. The advantage here is that it is much easier to remember orderly combinations of letters than it is to remember seemingly random numbers.

If the program setting up the higher level language is sufficiently sophisticated, it can handle the tasks of assigning memory locations and keeping up with these assignments. These tasks are time consuming and require close attention to detail when done manually.

Properly speaking, we should refer to this kind of program as an Assembler. While it can provide a lot of help to the programmer, it leaves a lot to be desired when it comes to simplicity. When we write a program with the assistance of an Assembler program, we must carefully give the computer very explicit instructions and pay a great deal of attention to registers, flags and accumulators. While we can write very full and complex programs with this system, we waste a lot of energy taking care of what amounts to bookkeeping dictated by the structure of the computer. We are forced to always keep track of where various bits of information are at any given moment and try to make sure that we don't lose one down in the depths of our computer.

Man's eternal quest for the easier way to do things is responsible for the development of the next higher levels of programming languages. Keep in mind that these languages are actually programs placed into the computer to allow the computer to help us write programs. The logical direction for higher level languages would be toward providing more assistance to the programmer. When you hear about languages such as BASIC, FORTRAN and COBOL, you are hearing the names of some of the higher level languages.

Though all of these languages are designed to make programming easier, each one places emphasis on different areas. Programs involving the use of large amounts of math are much easier to write in some languages than in others. Where word processing will be the primary area of application, another language would be a far better choice. As you might expect, some languages try to let you do a fair job in several areas.

If you manually multiplied 7.8 by 87, your results would look something like Fig. 1. You were able to arrive at the correct answer by using several sets of rules you learned in school. In essence, your teachers programmed you to keep track of decimal points, shifting partial products to the left and carries. If you are programming at the level of Machine Language or using a simple Assembler, you are forced to write out a similar set of rules in any program where you want the computer to be able to multiply.

Things would be much easier in a language such as BASIC. In BASIC, you would simply write something like 30 PRINT 7.8*87. This would cause the computer to print out the correct answer. Such simplicity using BASIC is possible due to two things. First, the computer and the programmer agree upon the meanings of the various commands and second, the program setting up the BASIC Language within the computer has a full set of rules covering multiplication.

```
   87
 x 7.8
  696
 609
 678.6
```

Fig. 1. Multiplication problem illustrating the different steps necessary to solve such a program in Machine Language or with an Assembler.

At this point, I'm sure that someone is wondering what the catch is. We all know that you don't get something for nothing but the way things are sounding, it would seem logical that if we were to write a program language at a high enough level, we could eventually tell the computer our problem and let it do everything else!

Perhaps this will happen someday, but for the present we're a long way from it. For one thing, the more complex we make the language, the more memory we need for our computer. Even a fairly simple language like BASIC can easily require more memory than many small systems have. Every step toward higher levels requires that much more.

Beyond this, we simply know very little about how the mind functions. The field of artificial intelligence is still wide open and there is a good chance that you could do some original work along these lines that would be of real value.

A "problem" . . . the analysis . . . a solution

At this point, I'd like to show how you as a programmer might analyze a problem in order to write a program. For many people, this step is the hardest and I want to show that there is nothing mysterious involved. Once this step has been completed, the programmer can then write out the program using the commands appropriate to the language he is using.

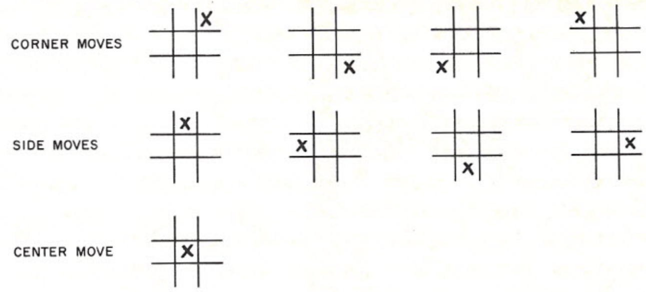


Fig. 2. The different moves encountered in tic-tac-toe.

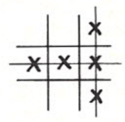


Fig. 3. SIDE wins.

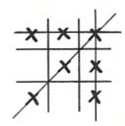


Fig. 4. CORNER wins.

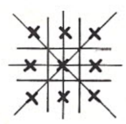


Fig. 5. CENTER wins.

The game of tic-tac-toe is well known and has been programmed many times. For our purposes, let's decide that we want the computer using our program to win when possible and avoid losing when it can't win. While this sounds obvious, we could just as easily decide to have the machine make random moves and be satisfied with an occasional win. The point is that we should always set out all the objectives of our programs as clearly as possible.

After we have decided upon our objectives, we need to start gathering information that relates to our problem. If you will refer to Fig. 2, you will see that we can refer to any possible move as either a side, corner or center move. Exploring all of the possible winning combinations for the above moves gives us some interesting data. Fig. 3 shows that if we occupy any side square, there are two possible ways to win with that square. Moving to a corner as in Fig. 4 gives us the possibility to win in three different directions. Finally, if we occupy the center square, there are four possible ways to win (see Fig. 5).

Based upon these observations, we can conclude that the center square is the most valuable square, that a corner square ranks second and that a side square offers a minimum of potential. These conclusions allow us to decide that we want to occupy the center square with our first move. If the computer's opponent moved first and took the center, we would then want to take a corner as they are the next most valuable type of square.

With this much information we can begin laying out a flowchart. Roughly speaking, a flowchart corresponds to a block diagram of an electronic circuit as opposed to the more complete (and complicated) schematic diagram. Its purpose is to show in simplified form how various functional units fit together. A particular block in an electronic block diagram may in reality represent a tube circuit, a transistor circuit or an integrated circuit. On a flowchart, any specific block will stand for any number of commands in whatever language is being used which will accomplish the function of that block.

A simple example of a flowchart can be seen in Fig. 6. It is based upon the first move our computer is to take. Start in block number one, and in this case, ask the question in the block. If the player wants to move first, the diagram indicates that we should move on to block number two and ignore block number three.

There is only one path out of block number two and following it takes us to block number four where again we ask a question. If the player did not take the center with his first move, we proceed to block number five which causes the computer to take the center. Had the player's first move been to the center, the chart indicates that we proceed from block number four to block number six. This particular block instructs the computer to take the second most important type of square, a corner.

A few moments spent going over the flowchart will show you that there are three valid paths through the different blocks. Two of these paths result in the computer taking the center, and in the third, the player takes the center and the computer responds by taking a corner. In all three cases, the computer finishes in block number seven which directs it to wait for the player to make a move.

Thus far, we've allowed the computer to make its first move and have it waiting for the player to move. One approach to completing our flowchart would be to develop a series of charts for each of the remaining moves. After finishing the chart for the first move, the computer would move on to the chart for the second move and continue from chart to chart until the end of the game.

While this approach will work, a few minutes spent studying the various combinations that can arise as player and computer complete successive moves will show that things get very complicated quite rapidly. In the interests of simplicity, we would be wise to seek another solution.

If we were to set up one general flowchart that would cover all of the possible situations, we could have the computer go through this one chart each time it made a move. While this flowchart would of necessity be quite detailed, its development isn't too difficult.

Essentially, we would start by having the computer determine if any possible move would allow it to win. If it found such a move, it would make that move and the game would be over. Failure to find itself a winning move would start the computer searching for a move that would block its opponent from winning on his next move.

If the computer didn't need to block the player from winning, we would direct it to search for a move that would allow it to win on its next move. Failing to find such a move, we would want the computer to see if it could keep the player from winning with two more moves. Should the computer be unable to make any of these moves, we would instruct it to make any move possible. If it couldn't make any move, we would then have it signal that the game had ended.

Fig. 7 shows the above steps in a simple flowchart. This flowchart is basically a closed loop with exit from the loop possible at only two points. Both of these exits cause the game to end. Close attention to this flowchart will reveal that there has been no provision made for the player winning a game. The reason is that if the principles shown in this and the first flowchart are followed, the player should be prevented from winning. Were this not so, failure to provide for the possibility of the player winning could cause the program to blow up if it ever encountered that situation. This is an important point because many program errors can be traced back to the failure of the programmer to take into account all of the possibilities the program will encounter.

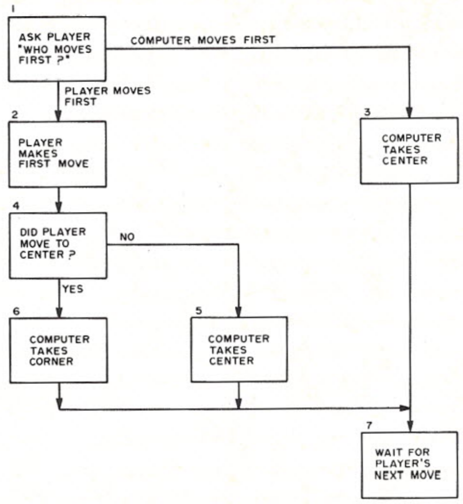


Fig. 6. Flowchart of first move.

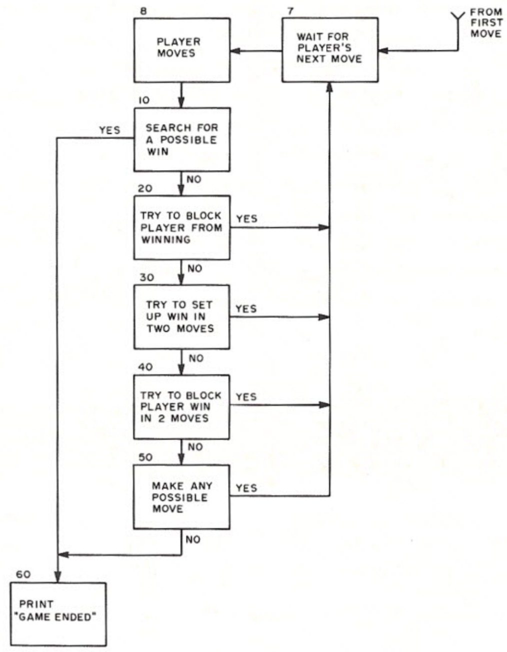


Fig. 7. Flowchart of all moves after first.

Fig. 7 should be considered a very broad outline. Its heart lies in the blocks numbered ten through fifty. While it is very useful in showing the general objectives of this phase of the program, we would normally want to develop more detailed flowcharts.

For example, block number thirty directs the computer to TRY TO SET UP A WIN IN TWO MOVES. What we would do is design a flowchart to meet this objective. Once completed, we would plug it into the position occupied by block number thirty. Similarly, we could develop charts for each of the other blocks and by combining all of these charts, produce a complete master flowchart.

We could have skipped making an intermediate flowchart like that shown in Fig. 7, but things are much easier to deal with if we take the time to do so. This kind of structured approach allows us to tackle problems of considerable magnitude by never allowing ourselves to be engulfed by the entire problem at one time. Instead, we first set up a general approach to the problem and then concentrate on small areas of the overall problem on a one-at-a-time basis. One important benefit of this method is the psychological reinforcement we receive as we successfully solve each part of the problem.

To illustrate how we might work out a detailed flowchart from one part of a general flowchart, refer back to Fig. 7. Block number ten is labeled SEARCH FOR A POSSIBLE WIN. While this statement is easily understood, it really covers a lot of territory. We are asking the computer to find a line of three squares where it already occupies two of those squares and the third is vacant.

In order to proceed, we need at this point to give every square some sort of identification. One way of doing this is shown in Fig. 8 where the squares are numbered one through nine. This allows us to have the computer check if it occupies squares one and two. If it does, we can have it determine if square number three is empty. If this were the case, we could then instruct the computer to take square number three and win the game.

Fig. 9 is a flowchart of the above sequence. If we had a series of such sequences covering all of the possible ways to win, we could develop a flowchart that would meet the objective of block number ten in Fig. 7. Before we do so, we should spend a few minutes trying to find out if there are any combinations of squares we don't need to examine. If we can eliminate any searching, we will be able to save execution time, conserve memory and simplify our program.

As it turns out, we can skip any steps which would result in the computer trying to take the center or square number five. The reason is that because of the way the first flowchart was handled, either the computer or the player must have already taken the center square. Therefore, it is impossible to make any move to the center and we can safely ignore those combinations which would require the computer to take it.

This reasoning allows us to save four searches. While the program would have worked had we kept them, this sort of tightening up is possible in most programs and can often make a critical difference in whether or not a program will run on our system.

Fig. 10 is a completed flowchart that meets the objective of SEARCH FOR A POSSIBLE WIN. While it appears complicated at first glance, it really isn't. What you see are twenty sequences of blocks like those in Fig. 9. The computer searches through the first set of blocks to see if it can win with that

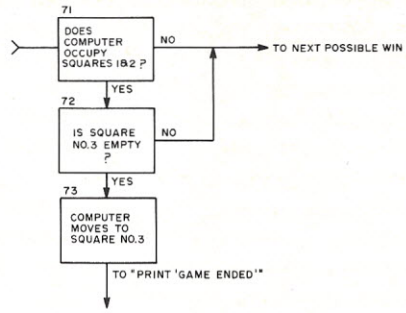


Fig. 9. Flowchart to check for one possible win.

1	2	3
4	5	6
7	8	9

Fig. 8. Numbering system for playing squares.

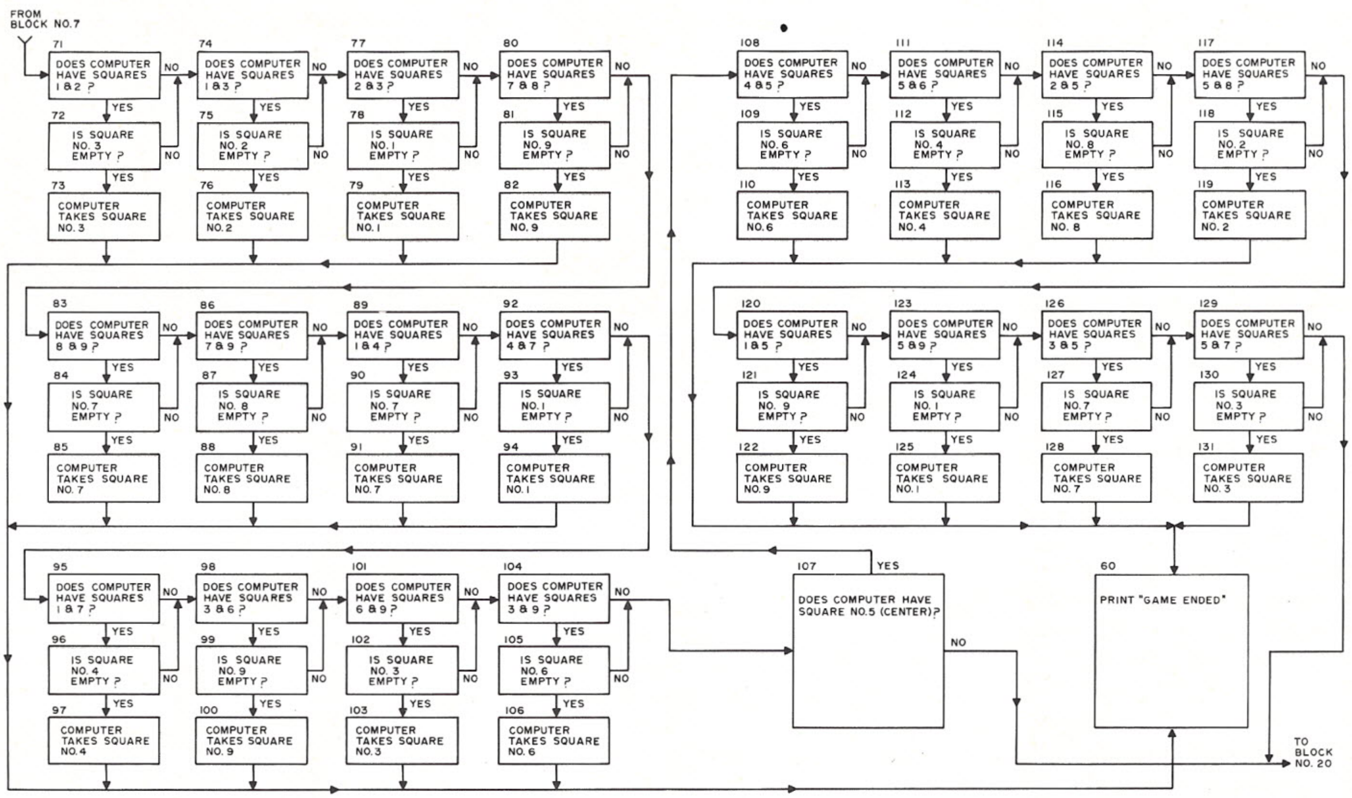


Fig. 10. Complete flowchart to "Search for a possible win."

combination of squares. If it can, it does so and prints GAME ENDED. If it cannot, it tries the next set of blocks to see if it can win there.

The computer will continue to test each succeeding set of blocks. Eventually, if it hasn't already found a win, it will arrive at a special block that asks the computer if it has moved to the center. This block could have been left out but its purpose is to save time. If the computer did take the center on its first move, this block directs it to proceed to those eight sets of blocks which test winning combinations based on already having the center. On the other hand, if the player occupys the center, there is no reason to go through those sets of blocks. Instead, the computer is directed around them and thereby saves the time it would take to make those eight searches.

It is something like trying to find a number in the phone book by either reading every entry starting on the first page or using a more systematic approach. Both methods will work, but the latter takes more rules to guide you to the correct entry. There are no strict rules about including this kind of time-saving step in your programs. What you are trading is extra memory to hold these steps against computer time spent going through needless operations.

At this point, you should have some idea of how to set up a flowchart. If you would like to complete one for the game of tic-tac-toe along the lines I have been following, let me make a few suggestions. Referring back to Fig. 7 will show that we have already designed a flowchart to meet the objective in block number ten. Block number twenty can be taken care of along very similar lines if we keep in mind that we want the computer to determine if the player has any two squares along any line where the third square is empty. If it finds such a condition, we want the computer to take the empty square.

Block number thirty can be handled by having the computer search for a line where the computer has one square and the remaining two are empty. A similar approach will work for block forty, and block number fifty involves searching for any empty square and taking it.

I am fully aware that computer programming is different and can sometimes be difficult. What I have tried to show in this article is that it isn't beyond the capabilities of the average person. What I haven't been able to get across is the deep sense of personal satisfaction that can be derived from personal programming. There is something about meeting a complex machine on its own terms and emerging its master that is really tremendous.

I hope that you are interested in finding out more about programming. If so, I strongly recommend that you either purchase your own computer or at least gain access to one at a school or business. Talk to people who already do their own programming and listen to their advice.

There are many books and courses available on computer programming and one of them should fit your particular needs. Just remember that reading about programs and programming won't make you a programmer. The trite and true fact is that you become a programmer by programming.

To help you set started, let me challenge you with a real problem. Try to draw up a complete flowchart to be used in calculating the total number of minutes any person would have lived from the time of their birth to the present minute. If you've already thought about Leap Years, you are fast. Have you thought about time zones? ■

Don Parks
1039 Ivy Lane
Cary NC 27511

Adding "Plop" to Your System

. . . a noisemaker for computer games

ADDRESS	HEX	MNEMONIC	NOTES
0200	A9	LDA, IMM	Output port
1	FF		Data
2	A2	LDX, IMM	First Half
3	00		of Timing Loop
4	8D	STA, abs	
5	00		Output port
6	FE		Address
7	CA	DEX	Loop-D-Loop . . .
8	D0	BNE	
9	FD		
A	A9	LDA, IMM	Output port
B	00		Data
C	8D	STA, abs	
D	00		Output port
E	FE		Address
F	A2	LDX, IMM	2nd Half of
0210	00		Timing Loop
1	CA	DEX	Loop-D-Li . . .
2	D0	BNE	
3	FD		
4	4C	JMP	Recycle
5	00		
6	02		

Program A.

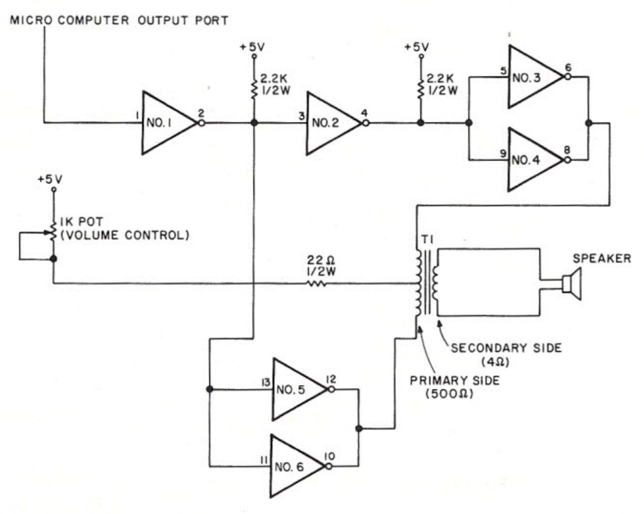


Fig. 1.

Here is a little noisemaking circuit that I've built into my last 3 home brew 6502 based machines. When attached to an output port and driven by a program loop, it will allow you to add sound effects to your programmed games.

Circuit

One inverter stage (#1) of the 7406 TTL hex inverter integrated circuit is used to buffer the output signal from the output port of the microcomputer (note: if buffering were not done, then the output port would have to drive 3 inverter stages which would exceed the limits of some I/O ports that were designed to only drive 1 TTL load).

A second inverter (#2) is used to invert the signal from inverter #1. This inversion is necessary so that the parallel inverters, #3 and #4, and #5 and #6 can be connected in what is known as a *push-pull* amplifier to the primary leads of transformer T1. (The inverters are in parallel to increase their power output.)

When the output port goes to a logic 1 (greater than +2 volts), the output of inverter #1 goes to a 0 (less than +0.4 volts) and the output of inverter #2 goes to a 1. This action causes the output of inverters #3 and #4 to go to a 0 and pull current (from the +5 volt power supply via the 1k Ohm volume control and the 22 Ohm current limiting resistor) through transformer T1 which causes the loudspeaker to make a single *plop* noise. When the output port goes to a 0, the outputs of inverters #5 and #6 go to a 0 and pull current through T1 (and again the loudspeaker goes *plop*).

If the output port is switched between a 0 and a 1 at a very fast rate, the signal heard in the loudspeaker will be a tone at the frequency at which the output port is switched.

The two 2.2k Ohm resistors (which could have been any value between 1k Ohms and 4k Ohms) are pull-up resistors to cause the outputs of inverters #1 and #2 to go higher than +2 volts when their outputs should be 1. The transformer and loudspeaker were pulled out of an old transistor radio.

Test Program

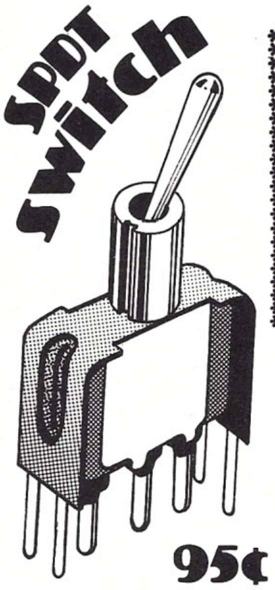
The 6502 program shown for *tootin' the horn* (Program A) is loaded starting at address 0200 and uses the address of FE00 for the I/O port (change this address to match your machine). The frequency of the tone is determined by the values at locations 0203 and 0210. Try different values in these locations – you will be able to vary the output tone from about 350 Hz to over 50 kHz (I used to work around jets – my hearing is completely gone above 40 kHz).

The actual bit in the output port to be switched is set at addresses 0201 and 020B (for convenience, I just turned all 8 bits on and off together). ■

GODBOUT

BILL GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614

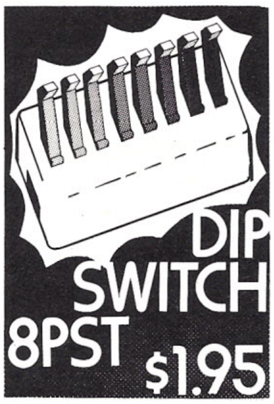
TERMS: Add 50¢ to orders under $10; over $10 add 5% for shipping & handling. BankAmericard®/Mastercharge®: call (415) 562-0636, 24 hours. Cal res add tax. No CODs.



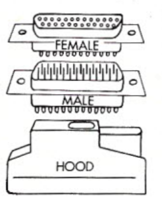
95¢

This beautiful, brand new SPDT miniature switch is designed for PC board mounting (bushing not threaded). Includes mounting frame for extra support.
PC MOUNT......ORDER #SW-3

ALSO AVAILABLE: Standard miniature SPDT toggle switch, w/ all mounting hardware. **$1.15** each or **5/$5.00**
STANDARD......ORDER #SW-2



LOOKING FOR A GOOD CONNECTION?



25 PIN RS-232 CONNECTORS: sub-mini D type.

Male plug with plastic hood, part #DB25P $3.95

Female jack, part #DB25S $3.95

74LS TTL

74LS00	$0.36	74LS139	1.38
74LS01	0.36	74LS155	1.38
74LS02	0.36	74LS157	1.25
74LS04	0.42	74LS160	1.85
74LS08	0.38	74LS161	1.85
74LS10	0.36	74LS162	1.85
74LS11	0.38	74LS163	1.85
74LS14	1.38	74LS168	1.87
74LS20	0.36	74LS169	1.87
74LS21	0.38	74LS174	1.38
74LS22	0.38	74LS175	1.35
74LS27	0.38	74LS221	1.38
74LS30	0.36	74LS240	1.88
74LS32	0.38	74LS257	1.25
74LS37	0.53	74LS258	1.38
74LS38	0.53	74LS273	2.25
74LS42	1.25	74LS283	1.20
74LS74	0.56	74LS367	1.00
74LS75	0.85	74LS368	1.00
74LS109	0.60	74LS377	1.88
74LS124	2.50	74LS378	1.38
74LS125	0.75	81LS95	1.13
74LS126	0.75	81LS96	1.13
74LS132	1.50	81LS97	1.13
74LS138	1.38	81LS98	1.13

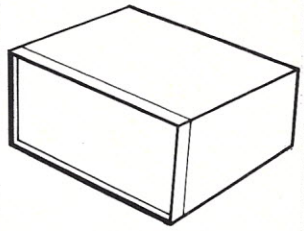
PLAY WITH WIRE?

SEND FOR FREE FLYER!

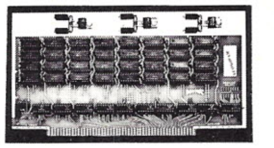
VECTOR-PAK enclosures

This is not another painted box, but is equivalent to the fancy enclosures used for industrial and computer oriented equipment. Tasteful matte color scheme and extensive use of brushed aluminum add to the beauty. It's shipped knocked down for easy drilling, machining, or labelling, and is easily assembled: now you can wrap your project up in style.

There are no screws or fasteners to mar the lines of these beautiful enclosures. Has provisions for card guides, connectors, etc. on the inside. Available in black or computer blue with white front panel; shipped unassembled.



#VP5-17-17U 5.51"H, 17.58"W, 17.1"D **$79.25**
#VP7-17-17U 7.26"H, 17.58"W, 17.1"D **$84.00**
#VP9-17-17U 9.01"H, 17.58"W, 21.6"D or exact same size as IMSAI microcomputer. **$96.50**

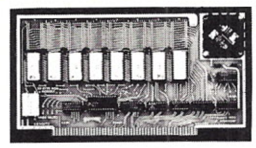


ECONORAM

KIT: $100 WIRED: $130

4K X 8, S-100 buss compatible full feature memory. Guaranteed faster than 500 ns, current draw under 750 mA...both specs over full temp range.

Fully buffered; DIPSWITCH address selector; sockets included for all ICs; uses lo-power Schottky support ICs: the best combination of value and economy.



ECONOROM

Our family of ECONOROM boards offers 3 different capacities of eraseable ROM... suitable for holding any program or routine you want. Similar to ECONORAM, and also very low power: 5V@½A, -12V@150 mA.

2K X 8 BOARD..............$135
4K X 8 BOARD..............$180
8K X 8 BOARD..............$270
4K 8080 SOFTWARE BOARD...$190

ALL PRICES ARE FOR KIT FORM

A BIG, BRIGHT .5" DIGIT

SHOWN ACTUAL SIZE

READOUT

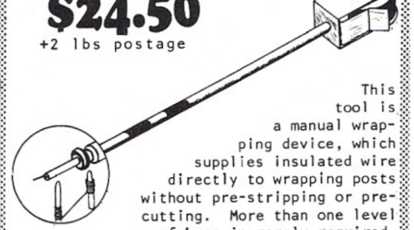
FND510 COM. ANODE...$0.95
FND503 COM. CATHODE $0.95
10 OF ONE KIND......$8.50

SPECIAL!

VECTOR'S "SLIT-N-WRAP"

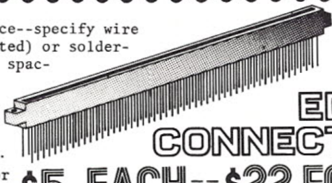
$24.50

+2 lbs postage



This tool is a manual wrapping device, which supplies insulated wire directly to wrapping posts without pre-stripping or pre-cutting. More than one level of wrap is rarely required. Speed your breadboarding time tremendously with this tool, and don't bother stocking all that pre-stripped and pre-cut wire you would normally require.

100 PIN EDGE CONNECTOR

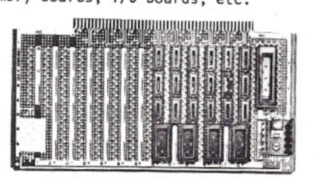


Now you have a choice--specify wire wrap pins (illustrated) or soldertail with .250" row spacing. For IMSAI and Altair peripherals. Wire wrap part # S-100WW. Soldertail part # S-100ST. #S-100ST is ideal for the IMSAI motherboard.

$5 EACH--$22 FOR 5

from VECTOR!

This is one of the neatest Altair accessories we've seen. It accepts virtually any size IC package, has a power and ground plane on opposite sides of the board for extra capacitance. Room for 4 regulators, 1 heat sink provided with board. By the way, the sockets are shown only for illustration, but they get the point across that you can stuff a lot of ICs on here --- implement your own memory boards, I/O boards, etc.



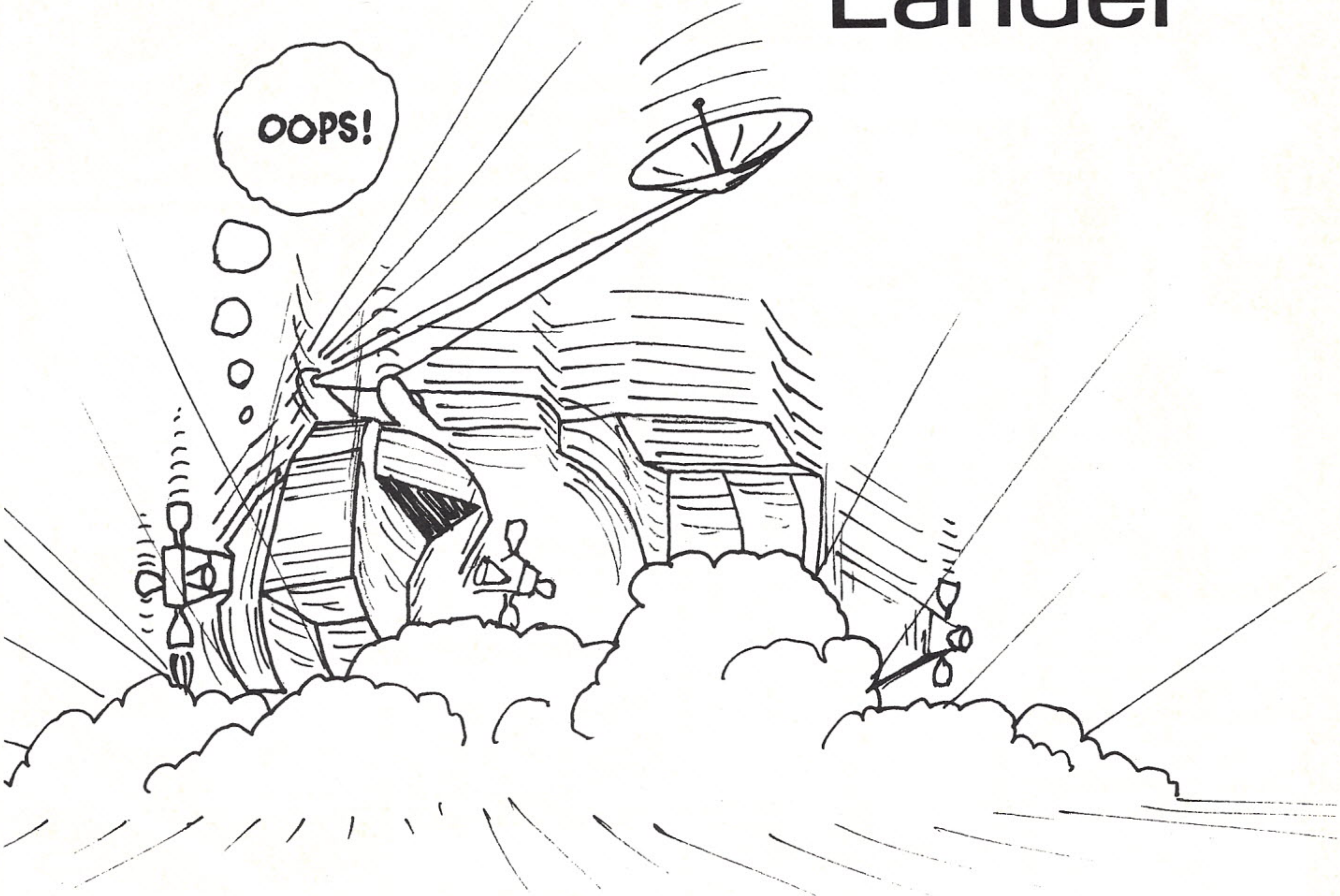
UNIVERSAL PROTOTYPE BOARD ➡ ➡ ➡ $19.95

LOW PROFILE sockets WIRE WRAP

Soldertail, tin plated	3 level, gold plated
14 pin........10/$1.95	14 pin........10/$3.70
16 pin........10/$2.15	16 pin........10/$3.85
18 pin........10/$2.75	18 pin.........1/$0.75
20 pin........10/$3.10	22 pin.........1/$1.50
22 pin........10/$3.50	24 pin.........1/$1.00
24 pin........10/$3.60	28 pin.........1/$1.25
36 pin........10/$5.50	36 pin.........1/$1.35
40 pin........10/$6.15	40 pin.........1/$1.75

Jim Huffman
Hufco
P.O. Box 357
Provo UT 84601

Lunar Lander




```
WOULD YOU LIKE INSTRUCTIONS 1 — YES 0 — NO? 1
YOU WILL BE GIVEN SOME FUEL & INITIAL CONDITIONS.
GRAVITY HAS A FORCE OF -5 FEET/SEC.
TO CANCEL GRAVITY — BURN 5.
                    HAPPY LANDINGS!

FUEL 120
SPEED -50
DISTANCE 500
ENTER YOUR BURN? 0

FUEL 120
SPEED -55
DISTANCE 448
ENTER YOUR BURN

FUEL 120
SPEED -55
DISTANCE 448
ENTER YOUR BURN? 5

FUEL 115
SPEED -55
DISTANCE 393
ENTER YOUR BURN? 25

FUEL 90
SPEED -35
DISTANCE 348
ENTER YOUR BURN? 95

IT'S ALL OVER BUT THE SHOUTIN' . . . GOOD BYEEEEEEEE
DO YOU WANT TO PLAY AGAIN? 0

READY

#
```

Program B

This program was generated because my neighbors had discovered that I had a computer in my basement, learned to play simple games such as Hi-Lo, and were in need of a more complex game. I took the idea from an article I read in *Popular Electronics Magazine* on the HP-25 programmable calculator. I rewrote the program for my HP-65, and, when I had MicroBASIC up and running, I rewrote the program for MicroBASIC.

This is a lunar lander program. The object of the game is to safely land your spacecraft on the surface of the moon. To begin with, you are given 120 units of fuel. You're given that the moon's gravity has a force of 5 ft per second, and your lander has a velocity of -50 ft per second. You are 500 ft from the surface of the moon. The object of the game is to make a soft landing, that is, touch the surface of the moon at zero speed. It is not easy. The program is written for MicroBASIC and uses only whole number arithmetic. It will work with no modification in full BASIC as long as any differences in operational characters are noted. The program listing is self-explanatory.

Fig. 1 is a flowchart of how the program operates, and Program B is a typical run. Program A is the complete program listing. ■

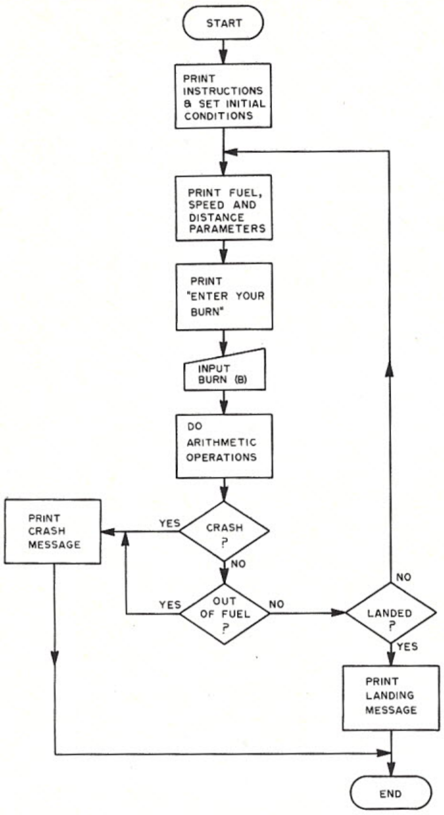


Fig. 1.

```
300   REM**** LUNAR LANDER
310   PRINT TAB (10); "LUNAR LANDER"
320   PRINT TAB (9); "++++++++++++++++++++"
325   PRINT
330   PRINT "WOULD YOU LIKE INSTRUCTIONS 1 — YES 0 — NO";
340   REM "YES = 1, NO = 0";
350   INPUT A
360   IF A = 0 GO TO 400
370   PRINT "YOU WILL BE GIVEN SOME FUEL & INITIAL CONDITIONS."
380   PRINT "GRAVITY HAS A FORCE OF -5 FEET/SEC."
390   PRINT "TO CANCEL GRAVITY — BURN 5."
400   PRINT TAB (8); "HAPPY LANDINGS!"
410   LET F = 120
420   LET V = -50
430   LET D = 500
440   PRINT "FUEL"; F
450   PRINT "SPEED"; V
455   PRINT "DISTANCE"; D
460   PRINT "ENTER YOUR BURN";
470   INPUT B
480   LET F = F-B
485   LET C = B-5
490   LET D = D + V + C/2
495   REM: C IS ACCELERATION
500   LET V = V + C
510   IF F <= 0 GO TO 610
520   IF V <= 0 GO TO 585
530   IF D <= 0 GO TO 610
535   GO TO 440
540   PRINT "DO YOU WANT TO PLAY AGAIN";
550   INPUT A
560   IF A = 0 END
570   GO TO 400
585   IF D > 0 GO TO 440
590   PRINT "CONGRATS! PERFECT LANDING"
600   GO TO 540
610   PRINT "IT'S ALL OVER BUT THE SHOUTING", "GOOD BYEEEEE"
620   GO TO 540
```

Program A

Silence Noisy Teletype Motors

... with a solid-state relay

Don Parks
1039 Ivy Lane
Cary NC 27511

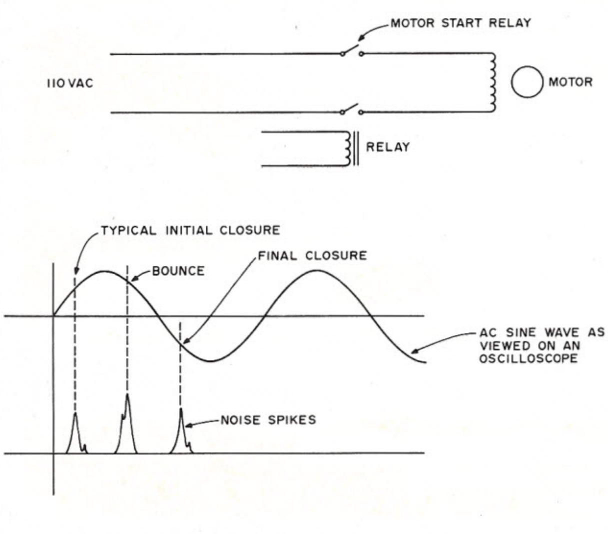


Fig. 1. As the motor started . . .

Once upon a time, back when 8008 microprocessor chips were selling for $125 each, even before Mr. Green started talking micros, I was trying to interface a homemade (out of old Teletype* parts) 5 level punch to my homemade 8008 system.

Oh, the anguish and grief!

Here's what was happening:

1) When the "start motor" command was received in the punch control unit, it pulled a relay and the motor started – at least that was what I'd had in mind.

2) What was really happening was this: (see Fig. 1).

a) As the motor start relay closed, the contacts bounced for 10 to 20 milliseconds,

b) the relay contacts caused a spark each time they closed or opened (when they closed, because of the current surge of the motor; when they opened, because of the discharge of energy stored in the motor coils),

c) noise pulses, about 20 volts peak, were observed (with an oscilloscope) riding on the +5 logic power supply.

d) The noise pulses were causing the TTL logic in the controller to change state which caused the relay to chatter – which caused more noise pulses – which caused the TTL – well, anyway, I think you get the picture.

Fig. 2 illustrates what I tried to do about the problem:

1) Put noise suppressors across the relay contacts (which helped a little).

2) Added bypass capacitors (shown in Fig. 3) to the power supply transformer's primary and at the output of the rectifiers and LM309K (5 volt) regulator – which, once again, helped a little.

3) I also added .02 uF, 25 volt capacitors next to each IC in the controller/interface logic.

After steps 1 through 3

*Registered trademark of Teletype Corp.

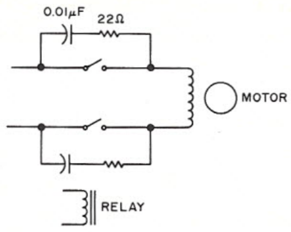


Fig. 2. Part of the fix.

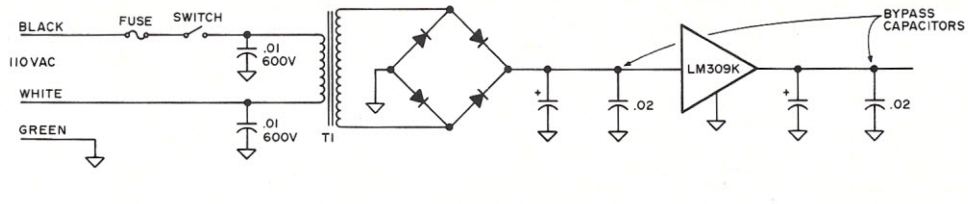


Fig. 3. More of the fix.

the punch almost worked right. Well, 2 out of 3 times, anyway. Fig. 4 illustrates what I used to finally fix the starting (and stopping) problem. Here's why it worked:

The solid state relay contains an optical isolator to make it TTL compatible, a "0" crossing detector (to determine when the ac sine wave crosses the "0" axis), and a TRIAC solid state switch (see Fig. 5).

When the +start motor signal goes to a logical 1 (greater than 2 volts), the output of the 7405 inverter goes to 0.4 volts and the solid state relay turns on at the next "0" axis crossing. When the +start motor signal goes to a logical 0 (less than 0.4 volts), the solid state relay turns off at the next "0" axis crossing.

Turning a circuit on or off at the "0" axis crossing of the ac sine wave is desirable because at that *instant* you switch 0 volts, hence no noise pulse. Contrast that with switching at the + or - peak of the sine wave when you get a *PEAK* noise pulse of about 160 volts. With an electromechanical relay you can neither sense the "0" crossing or reliably predict when (in the sine wave period) the relay contacts will first make contact.

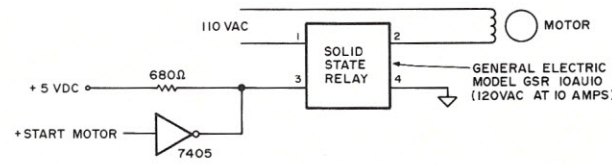


Fig. 4. The *fix.*

Turning on and off at the "0" axis crossings, plus the fact that the solid state relay has no contact bounce is what eliminated the electrical noise that was driving the TTL logic in the controller wild.

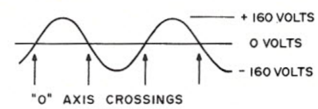


Fig. 5. Why it works so well.

Conclusion

Pay the $10 to $20 for a solid state relay when you must turn on fractional horsepower ac induction motors. It will save you a lot of grief (and tend to make you live happily everafter)! ■

Glen Charnock
864 Palomar Way
Oxnard CA 93030

Structured BASIC is Better!

Glen is a professional programmer, a ham and an avid computer hobbyist. He's got some interesting points here on applying structured programming techniques to BASIC (hey, I thought that couldn't be done). You might want to refer to Bill Jones' article on Structured programming in issue number 1 for additional information (flowchart symbols, definitions, etc.) — John.

Programming has traditionally been an art practiced by trained craftsmen. Structured Programming may well be the first step to changing programming from an art to a science. Structured programs are easier and faster to write, debug, or change. They are easier to understand and so may be more easily shared by users.

Structured Programming, or GOTO-less programming, began when a letter written by Edsger W. Dijkstra was published in *The Communications Of The ACM* in March, 1968. In his letter, titled "Go To Statement Considered Harmful," Dijkstra proposed that the GOTO statement caused more harm than good and should be eliminated from higher level languages (such as FORTRAN, BASIC, or COBOL). He went on to state that the GOTO statement was "too much an invitation to make a mess of one's program." Structured Programming has come to include many techniques in coding, standardization, and documentation. In fact it is difficult to get two people to agree just what structured programming does include. Many people will probably agree that the goal of structured programming is to improve programming through simplicity, standardization, and documentation.

Documentation

Little has been done to make documentation a science rather than an art so I will not say much about it. If you can give a friend a copy of your program and documentation materials and he is able to load the program and execute it properly, you are halfway there. Ask him how he would modify it to add a new feature or to change the function of an existing feature. If he can do this without further help, you probably have very good documentation. The larger the program, the more effort you should devote proportionally to good documentation. Don't be afraid to use remark statements in your programs (see Program C).

Standardization

Standardization has not made much progress either. It is difficult to get a group to agree on a standard for the features in a language let alone try to standardize how to write programs in that language. I will suggest some *standards* that I use, but if you have already developed a style which you use consistently, I would recommend against changing.

The most significant convention I use is a standard for line numbering. The first line in any code module (a section of code, 50 or fewer lines long, that performs a specific function) is numbered ending in 00. Following lines are numbered by tens, skipping any numbers ending in 00 since those are reserved for the first line of a module. The last line in a module is its exit line and has a line number ending in 99. The exit line is normally a RETURN, NEXT, or REM statement. See Program C.

It is hard to establish a standard for meaningful data names in BASIC since it only allows two characters in a name (any others are ignored). I try to avoid names using I or O since they are confused easily with one and zero. Try to make the names as meaningful as possible and try to relate any subscripts to the arrays they reference. For example, I recently had a program with arrays AV$ and AJ$ for adverbs and adjectives so I used AV and AJ to subscript each array.

I'll mention one more *standard* which can save a lot of time debugging. Never use an equals test in an IF statement if you can use a greater than or less than test instead. Most compilers and interpreters have little bugs in them waiting to trap the

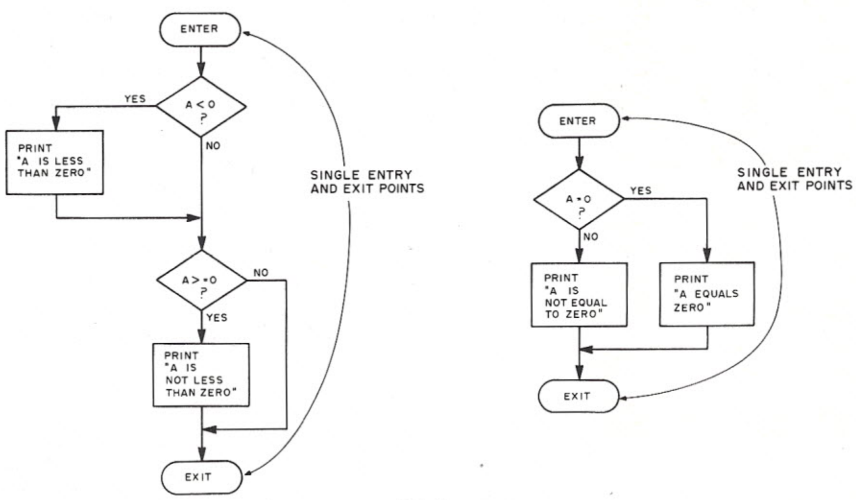


Fig. 1.

unwary. One common bug is a floating point routine which yields an innocent looking number but which has a least significant bit which will cause it to test unequal to the test number. In some cases a positive zero will test unequal to a negative zero so even testing equal to zero can yield wrong results.

Control Structures

Most of the progress in Structured Programming has been in methods of coding and simplifying the program's control structure. Two things have contributed to this advance: It was proven that GOTO statements are not required and may be harmful to a program, and it was recognized that the hardware prices were dropping rapidly while the programming costs were rising rapidly.

The changing ratio between software cost and hardware cost allowed a revolution in programming. Now programs can be written for people and not for the machine. The discovery of the GOTO's harmful role showed how to write programs for people. By writing more readable code the cost of software can be kept down. Even if it takes a few more machine cycles or an extra K of memory to use the new techniques, so what? Memory is cheap and the processor is fast.

In strict Structured Programming, GOTOs are completely forbidden. This is fine in theory but in practice it can lead to some very confusing code and in fact can cause more trouble than the GOTOs caused in the first place! In Russell Armstrong's book, *Modular Programming In COBOL*, he gives a set of restrictions on the use of GOTO statements in that language. We can change these restrictions a little and achieve a very practical approach to structured BASIC.

The first restriction is that GOTOs may only branch to a line with a greater line number than the line on which the GOTO appears. That is, they may only branch forward. This eliminates using GOTOs for looping so that all loops are controlled by FOR/NEXT loops.

The second restriction is that GOTOs may only branch to the first line of a module unless the source line and destination line are both within the same module. This guarantees that a module will have only one entry point as required by Structured Programming.

The third restriction is that GOTOs may not branch to a statement outside of the module containing the GOTO. This localizes the harmful effects of GOTO statements so that only those modules using them are likely to be confusing. It also guarantees that each module has only one exit point. The second and third restrictions allow us to rewrite any module in a structured manner since the module is guaranteed to have only one entry point and one exit.

The fourth and final restriction is that a GOTO may only branch to the exit of the module containing the GOTO. This restriction will eliminate the harmful effects of GOTO statements while still using them to minimize the need for excessively complicated IF statements. With this restriction in effect, whenever we see a GOTO we

```
10 IF B > A THEN GOTO 40
20 IF C > B THEN GOTO 80
30 GOTO 150
40 D = A
50 A = B
60 B = D
70 IF B > C THEN GOTO 150
80 D = B
90 B = C
100 C = D
110 IF A > B THEN GOTO 150
120 D = A
130 A = B
140 B = D
150 RETURN
```

Program A. Module to sort A, B and C so that A is the greatest and C the least. This is written in a non-structured manner.

```
100 REM SORT A, B, AND C SO THAT A IS MAX, C MIN
110 IF A < B THEN D = A: A = B: B = D
120 IF A < C THEN D = A: A = C: C = D
130 IF B < C THEN D = B: B = C: C = D
140 RETURN
```

Program B. Module to sort A, B and C written in a structured manner.

```
100 FOR L = 1 to 9999
110 REM THIS LOOP WILL REPEAT THE INPUT-PROCESS-OUTPUT
120 REM CODE UNTIL L IS SET TO 9999. IT MAY BE DESIRED
130 REM TO STOP EARLIER ON SOME CONDITION. IN THIS
140 REM CASE L MAY BE ASSIGNED THE VALUE OF 9999 IN
150 REM ORDER TO TERMINATE THE LOOP.
160 REM
170 REM GET INPUT
180 GOSUB 1000
190 REM
210 REM PROCESS INPUT
220 GOSUB 2000
230 REM
240 REM CHECK FOR END OF JOB AND IF NOT REACHED THEN
250 REM PRINT OUTPUT
260 IF L < 9999 THEN GOSUB 3000
299 NEXT L
```

Program C. This is a sample of the top level code for a simple program. Note that you can't have too many REM statements and that a blank REM line between sections makes the code easier to read. Also note the line numbering, especially the way line 200 is skipped — see text.

```
100 IF A < 0 THEN PRINT "A IS LESS THAN ZERO"
110 IF A > = 0 THEN PRINT "A IS NOT LESS THAN ZERO"
120 REM THE ABOVE CODE DOES NOT REQUIRE A GOTO
```

Example #1

```
130 REM
140 REM THE FOLLOWING SECTION DOES REQUIRE A GOTO
150 IF A = 0 THEN A = 1: PRINT "A EQUALS ZERO": GOTO 169
160 PRINT "A IS NOT EQUAL TO ZERO"
169 REM EXIT FROM IF SUBMODULE
170 REM
```

Example #2

Program D. The above code and flowcharts (Fig. 1) illustrate two ways of simulating an ELSE clause in BASIC.

know that the function of the module has been completed and we are ready to exit from the module and proceed to the next task.

Exceptions To The Rule

All rules have exceptions – except for this one. The same is true for our restrictions on the use of GOTOs. At least two significant exceptions should be mentioned: IF statements and error routines.

BASIC does not support the ELSE clause in an IF statement, so if we want to execute one of two functions depending on some condition, we have two choices. We can test for the condition being true and, if so, perform the first function; then we test for the condition being false and, if so, perform the second function. In some cases this will not work. The first function may affect the way in which the condition is evaluated when we test to see if it is false. For this situation we can test the condition, perform one function, then go to an "IF exit." Following the GOTO line, we code the other function and end this with the IF exit line. In effect we are using the IF code as a submodule with the IF as its entry and the "IF exit" as the module exit. See Program D and Fig. 1 for examples.

The second exception is the case where an error has been detected which is so severe that the program must terminate. In this case you may want to branch to a module which will print a diagnostic and halt the program.

Other exceptions may arise, just be sure that any GOTOs you add will help program clarity rather than causing confusion.

Top Down Programming

Top Down Programming is another set of ideas which have been grouped into the overall title of Structured Programming. The idea here is that you should write the *top* or most general level of code first and work your way down to the more detailed levels.

When writing a program, the first thing you should do is think about the problem to be solved. Determine the types of modules you will need to solve it. The first thing you will write then will be the *top* level which is usually a loop calling the logic routines needed to solve the problem. A sample of the top level code for a simple program is shown in Program C.

While this sort of technique may be cumbersome for a program to compute the square root of a number, it is quite effective when writing a payroll program or a new version of Star Trek. Even this is not strictly true. Program A shows a section of code designed to sort A, B, and C into descending sequence. This code was written in a nonstructured manner. The code in Program B performs the same function but was written by a structured programmer. Even for this simple a problem, the structured technique was able to make a significant improvement.

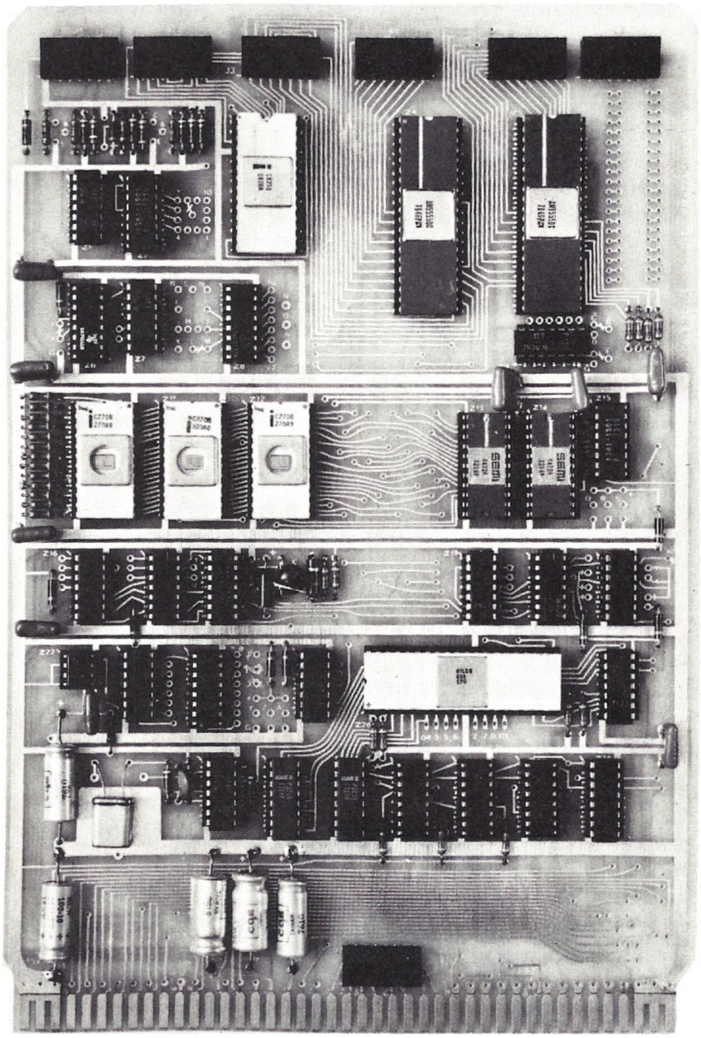
From my experience with structured programs I have developed a new corollary for Murphy's laws: Anytime you find that a program can't be conveniently written in a structured technique then you are not looking at the problem correctly and there exists a much simpler way of solving the problem. This may be simplified to: If it doesn't seem to work, you're doing it wrong. I have yet to find a program that can't be written more easily with structured techniques than without. Needless to say, I'm sold but you'll have to try Structured Programming for yourself to believe the difference it really makes. Give it a fair trial and you'll be glad you did. ■

SINGLE·BOARD or MULTI·BOARD Z·80 SYSTEM

NO SYSTEM COMPARES IN DOLLAR VALUE OR COMPACTNESS !!!

FEW CHALLENGE ITS CAPABILITY OR FLEXIBILITY

Our **RM Z-80 CPU SYSTEM** shown below, can be implemented as a stand-alone minicomputer system by merely adding a power supply and a Teletype or TV terminal as an I/O device (on-board interface included). ALONE, it stands as a fast, powerful minicomputer system. You can add memory or accessory boards at any time and when packaged in our RM Terminal case (below right), it is one of the most powerful self-contained desk-top minicomputers on the market. In addition to the CPU board, the case houses the power supply, keyboard, up to 64K of memory, a video display interface board, an optional Teletype, and special-purpose boards such as audio and digital cassette interfaces and a scientific calculator interface board. After all these, there is still room in the enclosure for your "home-brew" board built on one of several RM prototype boards available from MINI MICRO MART.



Shown above is a fully loaded RM Z-80 CPU System. Actual size: 7"x 10½".

Our **RM Z80-300 CPU SYSTEM** comes as a kit, complete with Z-80, TTL, resistors and capacitors, crystal clock, 1K of 255ns static memory and one 8255 interface IC, as well as 20-mil TTY circuitry. On-board provision exists for other parallel and serial interfaces as well as three 2708 E PROMs. All RM PC boards are epoxy glass, with plated-through holes and gold fingers. It is ....
INTRODUCTORY PRICED AT $199.95

If 1K operating monitor in a 2708 PROM is desired, order as **RM Z80-350** at **$264.95**

For all of above plus an additional RM 4K 300ns memory board, a 5-position backplane and two edge connectors, order as **RM Z80-S500** at **$419.95**

All of above and with surplus RM Terminal case shown below, which includes keyboard, power supply, fan, and card racks, order as **RM Z80-550** **$529.95**

All of above and with video display interface (no TV or monitor included), order as **RM Z80-650** **$629.95**

Power supply kit available; order as **PS-375** .... **$ 19.95**

4K static RAM boards available from $99.95 to $139.95.

16K static memory board available; order as **RM16K-300** **$479.95**

IC socket kits included with memory boards; add additional $14.95 for IC sockets for CPU board if desired. Order as RM Z80-LP.

Any board available assembled and tested for $50 additional. Write for quotation on complete systems assembled and tested.

Enclose $1 per board for shipping, handling and insurance; $10 additional for shipping RM Terminal or Teletype printer. (Excess shipping charges are refunded.)

UNTIL APRIL 30, 1977, ALL CPU BOARD PURCHASES INCLUDE Z-80 MANUAL, LISTINGS FOR MONITOR 80, AND 5K BASIC; PRICES SUBJECT TO INCREASE AFTER APRIL 30.

INTRODUCTORY OFFER -----
Audio Cassette Interface for above system --- runs BYTE (Kansas City) standard, HITS or Tone-No-Tone format and includes cassette tape of 3K Monitor 80 operating system and 5K BASIC, including manuals and source listings. Order as **RM-ARC-375** .... **$ 39.95**

Write for surplus catalog or S-100/Altair/IMSAI catalog. Send stamped, self-addressed envelope for specific catalog or other detailed information.

MiniMicroMart

1618 James Street, Syracuse, N.Y. 13203, Phone: (315) 422-4467

see next page for a listing of Altair Computer Centers

ALTAIR COMPUTER CENTERS

BEAVERTON, OR 97005
8105 SW Nimbus Ave.
(503)-644-2314

BERKELEY, CA 94710
1044 University Ave.
(415)-845-5300

SANTA MONICA, CA 90401
820 Broadway
(213)-451-0713

DENVER, CO 80211
2839 W. 44th Ave.
(303)-458-5444

ALBUQUERQUE, NM 87110
3120 San Mateo N.E.
(505)-883-8282;883-8283

TUCSON, AZ 85711
4941 East 29th St.
(602)-748-7363

LINCOLN, NB 68503
611 N. 27th St.
Suite 9
(402)-474-2800

LITTLE ROCK, AR 72206
2412 Broadway
(501)-371-0449

TULSA, OK 74135
5345 East Forty First St.
110 The Annex
(918)-664-4564

HOUSTON, TX 77036
5750 Bintliff Drive Suite 206
(713)-780-8981

RICHMOND, VA 23230
4503 West Broad St.
(804)-355-5773

SPRINGFIELD, VA 22150
6605A Backlick Rd.
(703)-569-1110

CHARLESTON, W. VA. 25301
Municipal Parking Building
Suite 5
(304)-345-1360

EAGAN, MN 55122
3938 Beau D'Rue Drive
(612)-452-2567

ANN ARBOR, MI 48104
310 East Washington Street
(313)-995-7616

WINDSOR LOCKS, CT 06096
63 South Main Street
(203)-627-0188

PARK RIDGE, IL 60068
517 Talcott Rd.
(312)-823-2388

ST. LOUIS, MO 63130
8123-25 Page Blvd.
(314)-427-6116

DAYTON, OHIO 45403
(513)-252-6785

BURLINGTON, MA 01803
120 Cambridge St.
(617)-272-8770

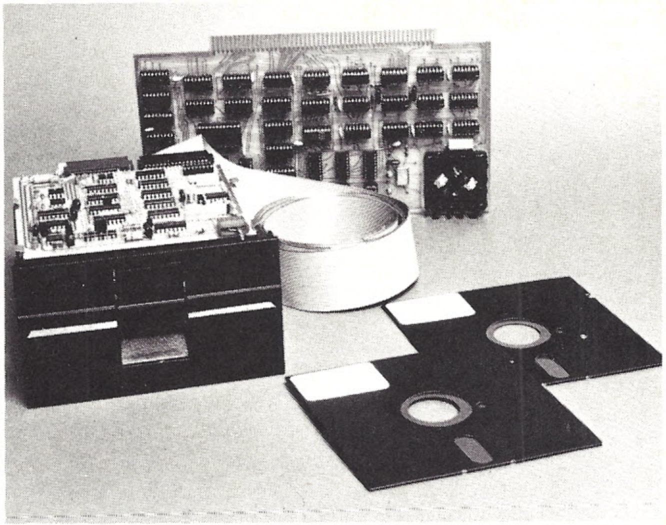
ALBANY, NY 12211
269 Osborne Road
(518)-459-6140

NEW YORK, NY 10018
55 West 39th St.
(212)-221-1404

ATLANTA, GA 30305
3330 Piedmont Road
(404)-231-1691

TAMPA, FL 33614
5405 B Southern Comfort Blvd.
(813)-886-9890

ASDC
3330 Peachtree Road, Suite 343
Atlanta, Georgia 30326



COMPLETE FLOPPY DISK SYSTEM FOR YOUR ALTAIR/IMSAI $699

That's right, complete.

The North Star MICRO-DISK SYSTEM™ uses the Shugart minifloppy™ disk drive. The controller is an S-100 compatible PC board with on-board PROM for bootstrap load. It can control up to three drives, either with or without interrupts. No DMA is required.

No system is complete without software: we provide the PROM bootstrap, a file-oriented disk operating system (2k bytes), and our powerful extended BASIC with sequential and random disk file accessing (10k bytes).

Each 5" diameter diskette has 90k data byte capacity. BASIC loads in less than 2 seconds. The drive itself can be mounted **inside** your computer, and use your **existing** power supply (.9 amp at 5V and 1.6 amp at 12V max). Or, if you prefer, we offer a power supply ($39) and enclosure ($39).

Sound unbelievable? See the North Star MICRO-DISK SYSTEM at your local computer store. For a high-performance BASIC computing system, all you need is an 8080 or Z80 computer, 16k of memory, a terminal, and the North Star MICRO-DISK SYSTEM. For additional performance, obtain up to a factor of ten increase in BASIC execution speed by also ordering the North Star hardware Floating Point Board (FPB-A). Use of the FPB-A also saves about 1k of memory by eliminating software arithmetic routines.

Included: North Star controller kit (highest quality PC board and components, sockets for all IC's, and power regulation for one drive), SA-400 drive (assembled and tested), cabling and connectors, 2 diskettes (one containing file DOS and BASIC), complete hardware and software documentation, and U.S. shipping.

MICRO-DISK SYSTEM	$699
(ASSEMBLED)	$799
ADDITIONAL DRIVES	$425 ea.
DISKETTES	$4.50 ea.
FPB-A	$359
(ASSEMBLED)	$499

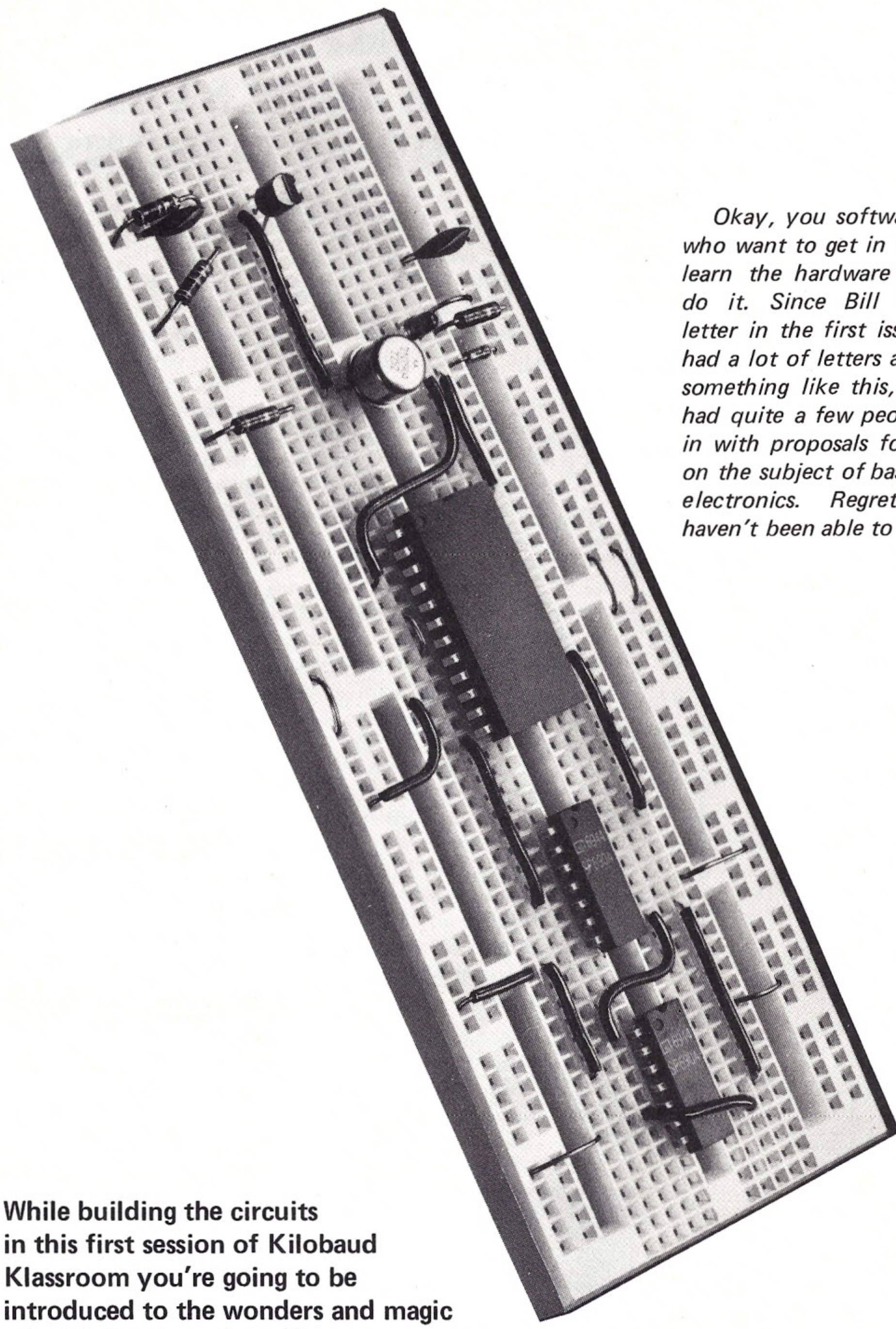
To place order, send check, money order or BA or MC card # with exp. date and signature. Uncertified checks require 6 weeks processing. Calif. residents add sales tax.

NORTH STAR COMPUTERS, INC.
2465 Fourth Street
Berkeley, CA 94710

N9

George Young
Sierra High School
Tollhouse CA 93667

Kilobaud Klassroom



While building the circuits in this first session of Kilobaud Klassroom you're going to be introduced to the wonders and magic of solderless breadboarding.

Photo courtesy AP Products, Inc.

Okay, you software types who want to get in there and learn the hardware . . . let's do it. Since Bill Pearson's letter in the first issue we've had a lot of letters asking for something like this, and I've had quite a few people write in with proposals for articles on the subject of basic digital electronics. Regrettably, I haven't been able to find anyone to give me what I want (and hopefully, you want) until George Young. All the others wanted to do articles of a straight tutorial nature. George takes a very practical approach and you'll find that in this series he doesn't throw out a bunch of electronic formulae and then try to show you how to apply them (which would be the conventional approach). Instead, he will lead you into actually building something and teach you what you need to know as you go along . . . and throw in the math and formulae if they're needed.

The long range objective of this series is to take you through an introduction to digital electronics, microprocessors, support chips and basic construction and design techniques to the point where you will be able to handle the final experiments . . . the building of a microcomputer system.

George is ah Electronics teacher at Sierra High School in Tollhouse, California, and aside from being somewhat unorthodox and dictatorial, believes that the simpler things are the better. Digital electronics should be kept simple . . . it's a lot easier to learn than other areas of electronics. — John.

. . . Part 1:

Getting the Ball Rolling

Anyone who tries to teach electronics via a nationally published, once-a-month magazine is nuts! Electronics is a very difficult subject to learn. Holding class once a month is crazy. How are you, the student, going to ask the instructor questions? Mail seems to be about the only way. And if I get too many letters (I'll answer every one of them) then next month's class session may not get written. So men, keep that in mind and temper your letter writing a bit or you may defeat what John, Wayne and I are trying to accomplish.

John says that I am dictatorial. Yes, that's a fair statement. I think that most teachers are. What I want you to do is *try my way first, then you can try it your way.*

We will learn our electronics a bit differently than you might expect. We will use the experimental approach. We will perform many simple (and some not so simple) experiments. Each experiment should work before you proceed to the next. With a once-a-month class meeting this should allow you time to complete each session.

I am going to make some assumptions.

1. You have no electronics background.
2. You desire to learn (you would not have read this far if this weren't true).

Even if you are an old pro in electronics, you will surely find some new twists to add to your bag of tricks.

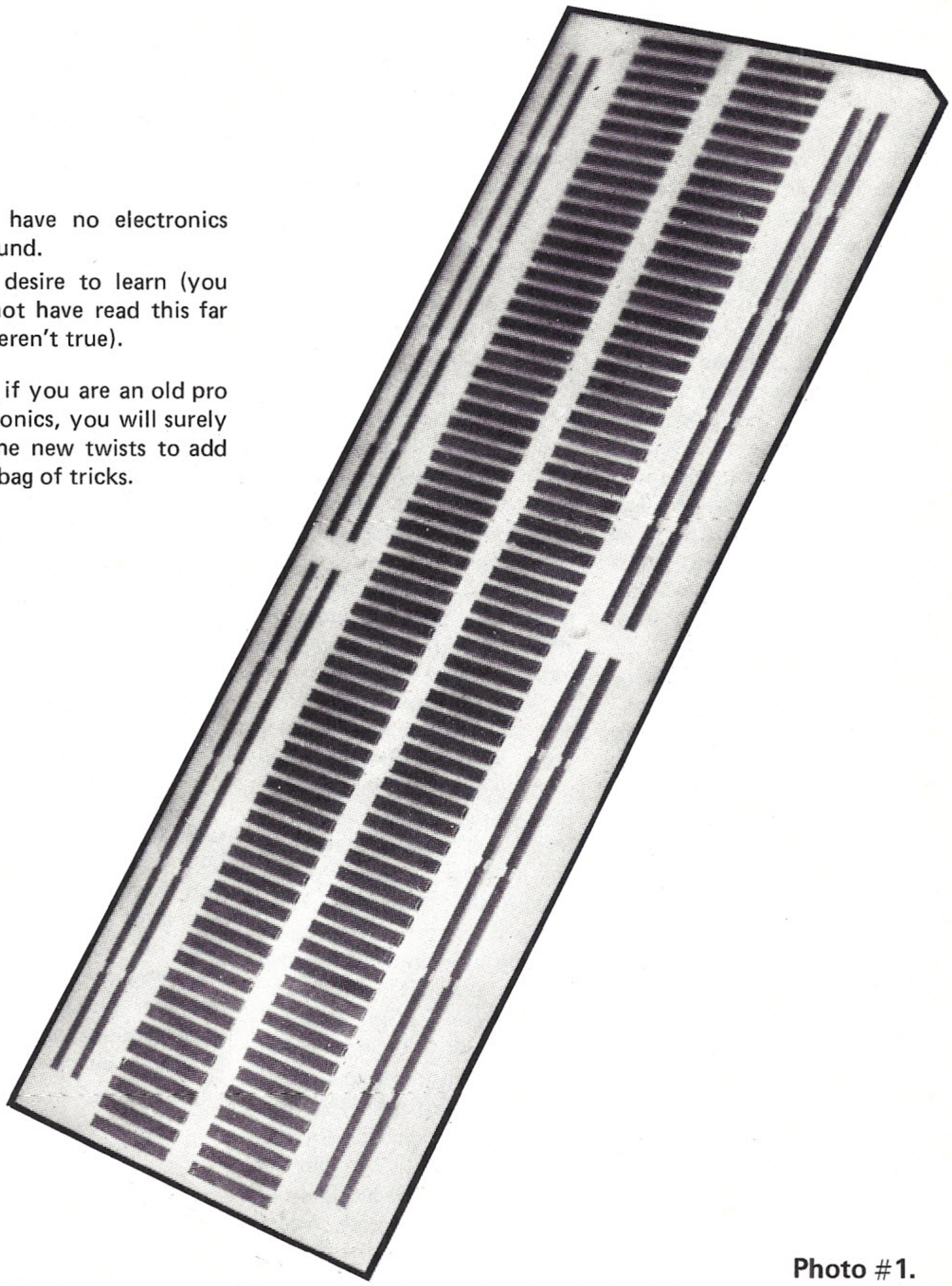


Photo #1. Superstrip with backing removed to show bus structure.

Overall Picture

We will start with the minimum that we can get by with. As the course progresses we will add to this minimum as our individual finances permit. We will build a console to do most of our experiments on. We will very quickly add a power supply because batteries cost dearly, even though we will start with them. We will teach you how to make a PC or printed circuit board. Our method will be a *one board method*, as inexpensive as we can make it. We will do the experiment first, then follow it up with the theory. The math will be held to an absolute minimum. We will include some tips on scrounging. We will even get some of us old-timers back to the local high school for some assistance. To help out a bit, we have arranged with another California firm to make up *experimental packages* of parts and supplies at a competitive cost to make it easier for you to get your needed materials. We have contacted some suppliers and arranged for a 10% discount for you if you mention Kilobaud Klassroom with your order. We will list two budgets which we will call *bare* and *loaded*. Adjust your buying accordingly.

We will use a plastic, plug-in, solderless breadboard. I recommend the AP Products, Inc. Superstrip. This can be ordered directly from AP Products, Inc., Box 110, Painesville, Ohio 44077, or you can use their toll-free number (800-321-9668) and they will tell you the name of your nearest AP Products distributor. You *must* mention Kilobaud Klassroom with your order, either to the factory in Ohio or to your nearest AP distributor to get your 10% discount.

In the budget list in Table 1 we have listed the other items that you must come up with for the first series of experiments. Some of them you will already have, while others will have to be obtained. A word about tools. A good tool costs money. It will last a lifetime. Cheap tools make tasks harder and do not last. Buy only quality tools.

Find the equipment list and get your parts and supplies on their way to you. While we are waiting for these items, I'm going to put you to work.

Building Our First Tool

Refer to Fig. 1. This is a probe. It is made from a defunct ball point pen. Remove the metal ink tube and store it. (We will use the metal tube later to make a wire-wrap tool out of this portion.) I will use a *Lindy* pen as an example. If the plastic shell is from a stubby, it is usable as is. If it is from a regular size pen, it should be shortened to about 8-10 cm. Pry off the plastic plug at the top and with a sharp knife, cut the plastic shell to a length of 10 cm. (So how come all this metric stuff? 1. Sooner or later you will have to learn it. 2. If some dictatorial character doesn't force you to, you'll go right on with the English system because it's easier to do so!)

Next, get a piece of stiff wire about 13 cm long. A wire coat hanger will do nicely. A piece of stainless steel welding rod about this same diameter is perfect.

We will assume that you are using the wire from the coat hanger. Remove the paint (coarse sandpaper or a file works well). Polish it bright and shiny. File one end to a long tapered point. Round the other end. Now "blue it." Heat it in a flame or even over an electric stove burner until it turns blue, then quench it in water. (I won't insult your intelligence by reminding you to hold it with a pair of pliers for this step.) Now glue the wire into the plastic shell. You can use any type of glue available to you for this purpose, but a hot glue gun is the fastest method.

What have you created? It's a probe. It's a scribe. It's a solder aid. It's a wire-wrap aid. It's a hole locator (center punch, but don't hit this one with a hammer). It opens holes on printed circuit boards that get plugged with solder. The rounded end burnishes letters in dry transfer lettering of panels. Girls can use it as a weapon to discourage rape. In fact, it is so handy, you had better make 3 or 4 of them while you're at it.

Parts Storage

You are going to need something to store parts in. If you are loaded, plastic storage trays in small cabinets can be used. If you aren't loaded, then try this method: Take nine empty milk cartons; pints, quarts, ½ gallons, it makes little difference. Wash them out and dry them. Cut off the tops so that all are the same height above the base (about 10 cm). Try and cut them off square.

Take three of them, and using masking tape or something similar, tape them in a group. Take three more and repeat. Repeat again for the last three. Now tape the group of nine together to make a tray. Different size cartons make different size storage trays. Repeat as required for parts storage. The cost is very little. The value is immeasurable.

Defunct Electronic Equipment

We are going to start what ham's call a junk box (some spell this "junque"). A junk box is a collection of parts that have been used before and have been salvaged for reuse. It can save money. It can mean the difference between building something you want and not building it because you can't afford it. Every ham radio operator who builds things has one, but not every ham is a builder. Citizen Band operators have one if they build, but not every CBer builds.

Start with a defunct piece of electronic equipment. Anything solid state will yield useful parts. Vacuum tube equipment will also yield some useful parts. The idea is to *salvage* everything that you can. *Don't* cut parts off boards or terminal strips. Unsolder them. Leave the wire leads as long as you can. It takes extra time, but it will pay off for you in the long run. Clean up the parts before you store them. Remove the excess solder. To remove excess solder, heat the part with a soldering iron until the solder is molten then rap the pliers sharply on the workbench. The solder will fly off and leave a clean lead behind. Hey, you guys doing this on the kitchen table, don't forget to cover the table with old newspaper or you will be sure to hear from the XYL. (An XYL is an ex-young lady, a term used by hams to refer to their spouse.) We will need controls, speakers, resistors, capacitors, and so forth. A power transformer will be needed later for the construction of a power supply.

The Art of Scrounging

The ham's junk box has already been mentioned. Locate a ham. It's easy. Walk, or drive down any residential street. Look for his antenna farm. You may contact a CB enthusiast instead of a ham, but some CBers also build. Introduce yourself. Tell the man or woman who you are. Get to know this person. The key word here is *swap*. As your own junk box builds,

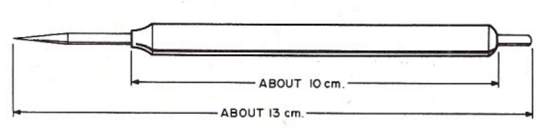


Fig. 1. Homemade probe.

you will build up swapping material. If you're a youngster, swap your labor.

I know many hams who have been promising the XYL to clean out that "mess" in the garage for months (sometimes years) now, and you might be just the fellow to lend a hand helping to cart off everything that is to be disposed of. Even if you don't swap anything at all, you will have gained a *resource person* with whom you can share some of your problems and you will have made a new friend.

Just a word of caution: Accept everything that is offered whether you have a use for it or not. You can always throw it away later if it is of no value to you.

Commercial operations: Every electronic repair shop has something for you. The problem here is production time. Every minute of any commercial operation is money. If you interrupt a workman, or the "boss," and take away valuable production time from them, you will take away nothing for your junk box. Offer to come back after working hours or on a Saturday. Leave your phone number and ask him to call you when it is convenient for you to return. There is a gold mine there if you can figure out the right timing.

Need some hook-up wire? Stop by the telephone repair truck the next time you see one. Ask the lineman if he has any scrap pieces of telephone installation wire that you can have. This wire contains 4 strands of about No. 22 or 24 wire in red, green, yellow and black and is exactly the right size for us to use in our experiments.

Your local school. As a taxpayer, you are part owner of your local school system. Call the school. Ask for the name of the electronics teacher. Look him up in the phone book. Call him. Make an appointment to go over to his house and meet him. He is another *resource person* for you. And the school itself may be able to contribute to your junk box. He will know the name of some local hams and can introduce you to them. What a shame that more taxpayers don't utilize this resource more often. One last comment on scrounging, and this is for the youngsters. Mind your manners and don't forget to say thanks! (Everything I've said about high schools also applies to junior colleges, even more so.) Hey, was that the bell? Time to get to class!

KILOBAUD KLASSROOM
Experiment No. 1
The Integrated Circuit Clock

The circuit used in this experiment is courtesy of Signetics Corporation. Reference: Signetics Data Manual, 1976, Microprocessor Section, Page 19.

Purpose:

1. To build something that works
2. To introduce some electronics concepts and symbols

Equipment:

1. Superstrip or equivalent breadboard
2. Hook-up wire
3. 7404 integrated circuit
4. Resistor between 150 Ohms and 470 Ohms, ¼ Watt (270 Ohms optimum)
5. Two electrolytic capacitors, 1.0 uF to 100 uF
6. Speaker: size and voice coil impedance not significant
7. Source of dc power: four dry cells satisfactory

New symbols introduced: Since this is the first experiment, all the symbols will be considered new. Refer to Fig. 2 while we discuss these new symbols.

The Inverter — The first experiment will utilize the 7404 Integrated Circuit (IC). The 14 pin Dual In Line (DIP) package contains 6 separate inverter circuits shown in Fig. 2a along with the Pin Outs (P.O.). Note that the inverter symbol can be drawn in two different ways (with the circle at the input or output). The logic function is still the same — inversion. We'll discuss the difference later. For now, just accept it.

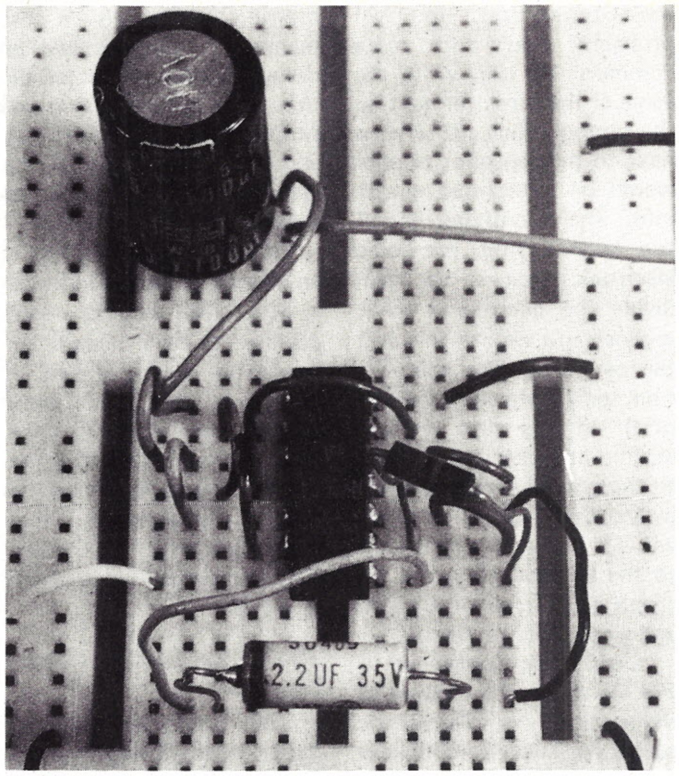


Photo #2. Experiment #1 set up on the Superstrip.

The Resistor — The symbol for a resistor is shown in Fig. 2b. We will write the value of the resistor next to the symbol. For example, 270 next to a resistor symbol would indicate a value of 270 Ohms. The unit of resistance is the Ohm. The abbreviation for Ohms is omega (Ω). 270 Ohms = 270Ω.

Capacitor — Refer to Fig. 2c. The symbol for the capacitor may be written in two ways. Either way is equivalent. The unit of capacitance is the farad. This unit is too large for electronic work. We divide the farad into 1 million parts. One of these parts is called a microfarad. The abbreviation for microfarad is mfd (or μFd or uF). Since the greek letter μ is not found on

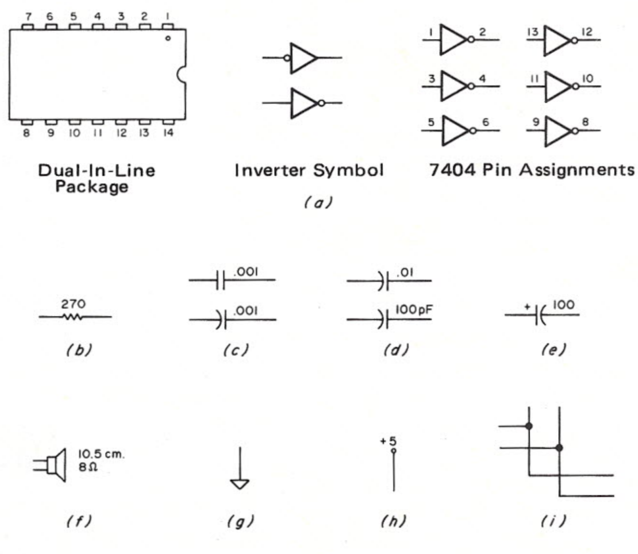


Fig. 2. Electronic symbols for Experiment #1.

most typewriters, you will probably find mfd more common. We will write the value of the capacitor next to the symbol. Thus, .01 written next to the capacitor symbol would be interpreted as .01 mfd. The microfarad is still too large a unit for some electronic work, so we again divide the microfarad by 1 million, and we call this unit the picofarad. The abbreviation for picofarad is pF (or pFd). In this class we will assume the value written next to the symbol is in mfd. If we intend the value to be in pF we will make note of it next to the symbol. Thus, Fig. 2d shows a capacitor of .01 mfd and a capacitor of 100 pF.

Electrolytic capacitor – Fig. 2e shows the method of designating an electrolytic capacitor. The + symbol next to the capacitor symbol signifies a polarized capacitor. A polarized electronic part is one that must be inserted into a circuit in one direction only. The + on the part must be aligned with the + in the circuit for proper component operation.

Speaker – The symbol for a speaker is given in Fig. 2f. As this symbol is drawn, it indicates a Permanent Magnet (P.M.) speaker. If the diameter and/or the voice coil impedance is important for the particular circuit application, these values would be written next to the symbol. Thus, Fig. 2f indicates a 10.5 cm diameter, 8 Ohm voice coil speaker.

Circuit Common – The symbol illustrated in Fig. 2g is common. Until you are informed otherwise, it will mean the negative (-) terminal of our power source.

Positive power source – Refer to Fig. 2h. This is the method that will generally be used to indicate the voltage of our power source. We will always attempt to draw the circuit in the simplest manner. A wire could be drawn throughout the entire circuit diagram connecting each point to power in sequence. If this produces a simpler drawing, then this is the manner in which it will be done. However, most of the time it is less confusing and simpler to use the small circle and write the value of the applied power next to the symbol.

Wiring – A straight line in an electronic circuit indicates a wire. This wire is always assumed to be insulated. The insulation is not drawn. If two wires cross each other but are not connected, there is no dot joining them. If two wires connect, a dot on the diagram indicates that they are electrically connected. This convention is illustrated in Fig. 2i.

The Experiment Circuit Diagram

The circuit diagram for Experiment 1 is shown in Fig. 3. In this first experiment, we will go into some detail on how to use the Superstrip. The photograph at the beginning of the article shows the Superstrip with some chips plugged into it. Photo #1 shows the back side of a Superstrip from which the adhesive backing has been removed in order for you to observe the internal bus structure of the Superstrip. Note that the power distribution buses running down each side have an opening in the center. It is necessary to place wire jumpers in the center of the Superstrip in order that we have continuous power distribution down each bus along the edges.

The 7404 IC (which is the heart of experiment #1) plugs in, straddling the center channel. This leaves 4 holes visible on each side of each pin on the IC. Refer to Photo #2. These 4 holes allow 4 separate connections to be made to each pin of the 7404. Orient the 7404 so that the identification notch or dot faces you, and you have pin 1 and 14 closest to you.

The first connections you will make are power and ground to the IC (or ICs if there are several). Always make the ground connection first, then the power connection, and always make these two connections before you make any other circuit connections.

Remember this – No circuit will function without proper power being applied. Get in the habit of connecting ground and power to the ICs now at the very beginning. The very first time that you troubleshoot a nonfunctioning circuit and discover that this is the problem, you will be kicking yourself all over the room!

Referring to Fig. 3 and Photo #2, the next step is to add the wire jumpers. After the jumpers are in, we add the components; resistors first and capacitors and diodes (if any) last. Two reasons for this: a) Bulkiest components get added last in construction, and b) polarized components get added last in construction. Finally, the moment of truth has arrived. Will it work? Connect a source of power (5 volts) to power and ground. Outside bus or rail is +5, inside rail is (common). Your creation should "squeal" at you. If you do not have a source of power available, you'll have to go on to Experiment 2 so we can find out how to get some power for this turkey.

Theory – Refer to Fig. 4 during the discussion of how this thing works. Fig. 4a shows the resistor connected from pins 1 & 2 of the first inverter element in the package. This resistor changes this inverter from a digital circuit to an amplifier. The circuit has been made to function with resistance values between 150 Ohms and 470 Ohms, with a value of 220 or 270 Ohms proving to be about optimum. If you do not have a 270 Ohm resistor (red-violet-brown), then try what you do have: 150 Ohms (brn-grn-brn), 22 Ohms (red-red-brn), 390 Ohms (or-wh-brn), 180 Ohms (brn-gry-brn), 330 Ohms (or-or-brn), or 470 Ohms (yel-v-brn). Fig. 4b shows the second inverter section, pins 3 & 4. This section inverts, or turns upside down, whatever it finds coming in on pin 3. The inverted signal is then fed back via the capacitor to pin 1. The phase is such as to cause the combination to oscillate. We will go into phase and oscillations later in the course. An oscillator generates an alternating current signal. In this case,

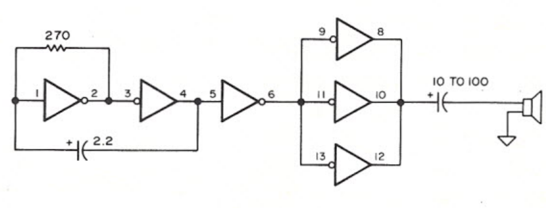


Fig. 3. Circuit diagram for Experiment #1.

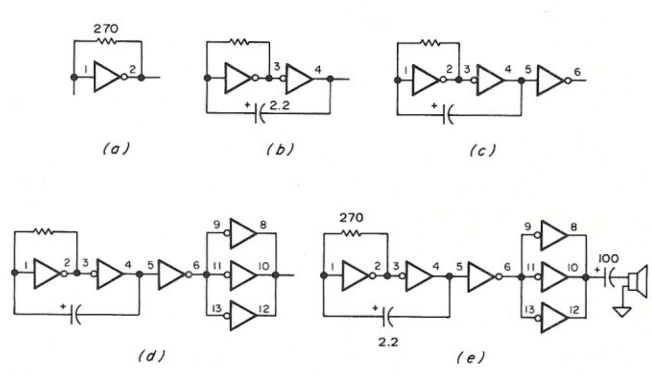


Fig. 4. Circuit theory details.

because we are using digital logic elements we get pulses of energy. A circuit that generates this kind of electrical pulse is called a clock.

Fig. 4c shows the third inverter section, pins 4 & 5. This inverter section also inverts the clock signal, but it also acts as a buffer. A buffer is an amplifier or digital stage that is used to separate one portion of an electrical circuit from other portions. Here we are using this section as a buffer to separate our clock generator from the amplifier stages which will follow.

Fig. 4d shows three inverter sections connected in parallel. All 3 input pins are connected together, and all three output pins are connected together. Connected in this fashion, the three sections will supply more current to the speaker, boosting the speaker's output.

Fig. 4e shows the coupling capacitor and the speaker. The capacitor couples energy from the booster stage to the speaker. The speaker is a transducer that changes electrical energy to sound energy. The larger you make this capacitor, the louder the sound. However, if the capacitor is omitted, the wire that forms the voice coil of the speaker will look like a short circuit to the booster and you will hear almost no sound out of the speaker.

To investigate the action of the booster, disconnect 2 of the 3 inverter stages. (Pull the wires out that connect pins 9 & 13 to pin 11, and pins 8 & 12 to pin 10. Leave pins 10 & 11 connected.) The level of the sound should decrease. Now add in just one more inverter section. (Reconnect pin 9 to 11 and 8 to 10.) You should hear an increase in volume. Now add the third stage by reconnecting pins 13 to 11 and 12 to 10. It didn't make as much of a change, did it? See if you can give a logical explanation for this.

I need to reemphasize a definition — a transducer is any device that changes one form of energy to another.

KILOBAUD KLASSROOM

Experiment #2

A Source of Power for Our Experiments

Purpose:

1. To provide a source of power for experiment #1.

2. To introduce diodes and batteries.

Equipment: (Only additional equipment needed will be listed.)

1. Silicon diode, 200 mA or more, 50 PIV or more.

2. 4 flashlight cells or a 6 volt lantern battery or 6 volt "hotshot" battery.

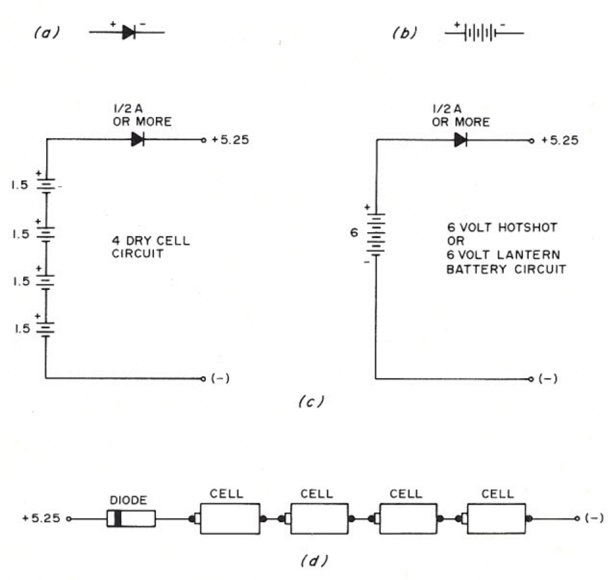


Fig. 5. Circuit symbols and diagrams for Experiment #2.

New Symbols

Diodes — Fig. 5a shows the symbol for a diode. A diode is polarized, which means it must be installed in a circuit in one direction only. The positive and negative ends of the diode are indicated by the plus and minus signs shown with the symbol (which are not normally included as part of the symbol). More on *how* it works later.

We will indicate the minimum desirable current next to the diode symbol. Thus, 1 A next to the symbol would indicate a current rating of 1.0 Ampere. The Ampere is the unit of electrical current. It is a measure of the amount of current a circuit is using (i.e., taking from the power source). This is often too large and so we divide this unit into 1000 parts which we call a milliampere and abbreviate mA. Again this is too large at times, so we divide this again by 1000 and call this unit a microampere and use the greek letter mu so that the abbreviation looks like this: microampere = μA.

Batteries — Fig. 5b shows the symbol for a battery. A single cell is sometimes drawn with just a long and a short line, but this can be misinterpreted as a capacitor symbol, so we will always use more than just one pair of lines, and we will indicate the positive end of the battery or batteries with a + sign.

Circuit

Fig. 5c shows the circuits that we will use. If you use 4 flashlight cells, they must be connected in series so that the voltages will add together. Each dry cell produces 1½ volts, so 4 of them connected in series will produce 4 x 1½, or 6 volts. A six volt lantern battery or a six volt "hotshot" battery is nothing more than 4 individual 1½ volt cells connected in series and packaged together. The two small circles in the symbols represent the points of connection to the battery.

Procedure

Connect up your choice according to the appropriate circuit diagram. The - end of the diode is indicated by a band or by a - symbol. Fig. 5d shows what is called a wiring diagram (as opposed to a circuit diagram or schematic diagram) and illustrates in detail how 4 dry cells would be connected in series with the diode in a pictorial form.

Theory

The ICs that we are using are designed to operate at 5.0 volts. The 6.0 volts produced by our battery supply is outside the allowable voltage range recommended by IC manufacturers. We could use a resistor to drop the voltage, but that would introduce some things that we are not yet ready to tackle. A diode connected with its positive end (made out of "P" material — sometimes called the anode) connected to a positive voltage is said to be forward biased. When a diode is forward biased, it will conduct. If the diode is connected with its negative end (made out of "N" material — sometimes called the cathode) toward + on the power source, it is said to be reverse biased and it will not conduct. Thus, a diode is a device that allows electrons (current) to flow through it in one direction only, and it will allow them to pass through only if the diode is forward biased. We have connected the diode here so that it is forward biased. Now forward biased diodes (and transistors) exhibit a very interesting effect when they are forward biased. They exhibit a voltage difference, or voltage drop, between the two ends of the diode. This voltage difference, or drop, is characteristic of the material from which the diode is made. Most modern diodes are made from silicon or germanium. Silicon diodes will exhibit a drop of about 0.6 volt when forward biased, while germanium diodes will measure about 0.2 volts. These voltages are approximate. We will say that silicon has a 0.75 volt drop and germanium a 0.25 volt drop. This is close enough to the actual case and seems to be easier to remember. Note that one voltage is about three times the other.

What I've just told you is extremely important. With a voltmeter and a diode (or transistor) connected so that it is forward biased, you can tell if the device is made from silicon or germanium, and you can identify the - or cathode end of a diode even if it is not marked. So, by adding a forward biased silicon diode to our six volt power supply, we can reduce the voltage from 6.0 volts to about 5.25 volts and this voltage is within the range for operating our ICs.

By now your experiment #1 either squawks at you or it doesn't. If it squawks at you, try changing the value of the resistor between pins 1 and 2 to note the effect on the pitch. Then try changing the feedback capacitor to a different value and note its effect.

A word of caution: When changing things around in a circuit that is operating (and by that I mean one that is in working condition — not one with power applied) change only *one* thing at a time. If you change more than one thing, you will not be able to evaluate the effect and you may even end up with a nonfunctioning circuit.

Troubleshooting

Suppose it doesn't work. Would you believe me if I tell you "great." Now you will learn even more. I didn't think so. Well, its true, and before we finish the course, I'll try and get you to change your mind.

We will present here a rather lengthy discussion on troubleshooting procedure. Some readers will not read this section until later, when they do have a problem, but every student will get back here sooner or later.

1. First, if you've been working on this circuit for an hour or more, then stop right now. Turn things off and go watch the telly for awhile. Go talk to the XYL. Get away from the circuit for a time.

2. Second fundamental: Proceed logically and slowly. You'll actually get things going sooner if you take your time and don't rush.

3. Check your wiring. A single wiring error will always prevent operation. (The opposite is, however, not necessarily true. Correct wiring does not guarantee an operational circuit.) Heathkit suggests going over each line in the circuit diagram with a colored pencil as you check the wiring.

4. Double-check your wiring for power and ground to the IC. (I told you to get in the habit of making these two connections first.) This is by far the most common error that my own high school students make.

5. Wet your finger with a little saliva. Touch the top of the chip. If you get steam, disconnect power quickly. If you don't get steam, does the chip feel warm? It should if it is getting power. This is a very simple test, but it is amazing how many technicians don't think of it.

6. Remove the IC from the Superstrip. Examine the pins. Did any of them get bent over upon insertion? This has happened to me more times than I care to admit. It is all too easy to bend over one pin when inserting a chip and this is almost impossible to detect from the top side of the IC.

7. Up to this point I have assumed that you do not have a voltmeter or a logic probe available for use. A voltmeter is absolutely necessary for our work, so plan now to get one as soon as you can. Heathkit has about the best prices and their kits are fun to build. Get a good meter. It is a lifetime investment and a good one will pay for its additional cost many times over. If you have a voltmeter and know how to use it, we can proceed to measure some voltages. I will eventually get to this, but we cannot do everything at once. We will also build a simple logic probe in the next couple of experiments to get us by for awhile.

Connect the ground chip to minus on the power source. With the probe, measure the voltage on pin 14 of the IC. When measuring voltages on a chip, measure the voltage on the pin at the point where the pin itself enters the plastic package. If you measure anywhere else and a pin is bent over and not making contact, you will be misleading yourself. If you measure 5 volts at pin 14, you are only half done. Next connect the probe to the positive power source terminal, and with the ground clip, measure the voltage again on pin 7. (Or simply place the ground clip directly on pin 7 and the probe directly on pin 14.)

8. At this point we have reached another fundamental point in our troubleshooting. This second basic idea is that of dividing a circuit in half. Right now you should be prepared to say whether the trouble is in the power supply half or in the circuit half. The concept of localizing a problem to ½ of a circuit is really the foundation for all troubleshooting.

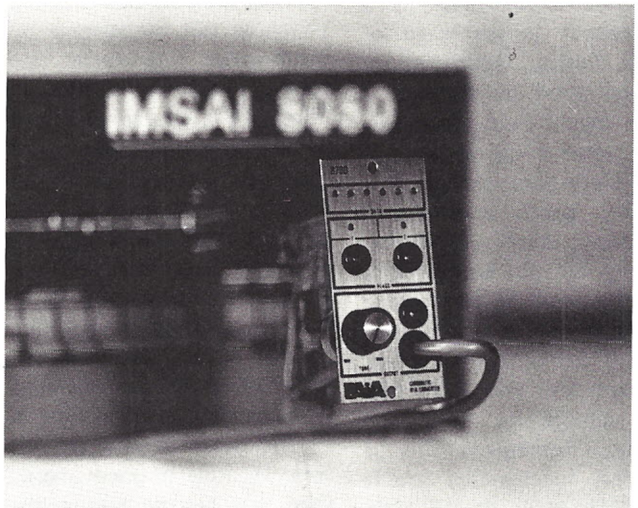
9. Let's assume for a minute that you have trouble in both halves of the circuit. Which half should you tackle first? If you think for a minute, you'll have the answer. And this brings us back to a troubleshooting fundamental mentioned earlier. *No electronic circuit will function if it does not have the correct power*. So we have to fix the power supply half of the circuit first. We now find the cold solder joint, or the defective cell, or the diode we put in backwards and get the power supply delivering +5 volts. Now we can turn to the other half of the circuit.

10. Can we logically divide this circuit in half (so that we are really looking at only ¼ of the circuit now, instead of at the entire thing)? Yes, the clock generator forms ½ and the booster-output circuit forms ½. Which half do we tackle first? Hint: If the speaker doesn't work, we will never hear the clock generator. How can we test the speaker? Let's assume that we don't have anything else available except what is in front of us. We've got a power supply working. But do not connect the speaker directly across the power supply terminals or we will have smoke and a bad speaker for sure. The voice coil of the speaker is a wire. If we connect it directly across our power supply, we will have a short circuit and something will have to give. It will be the speaker for certain.

However, if we place a resistor in series with the power supply, so that the current must also pass through the resistor as well as the speaker, then we can hear a click from the speaker if it is functional. So grab a resistor from your stock pile and connect it in series with the speaker and power supply but just touch the lead of one speaker terminal very quickly. If you get a click, you have a good speaker. If you don't, the resistor selected may be too big. Try a smaller one. Keep trying smaller ones and listening for a click. If you try every resistor that you have available and still don't have a click, then, and only then, quickly touch the power supply leads directly to the speaker terminals without a resistor. If you still don't get a click, then you have proven a bad speaker and you'll have to get another one. If you had a second speaker all this time, a better approach would have been to substitute the alter-

OUR COMPUTER MAKES MUSIC!

HERE'S HOW:

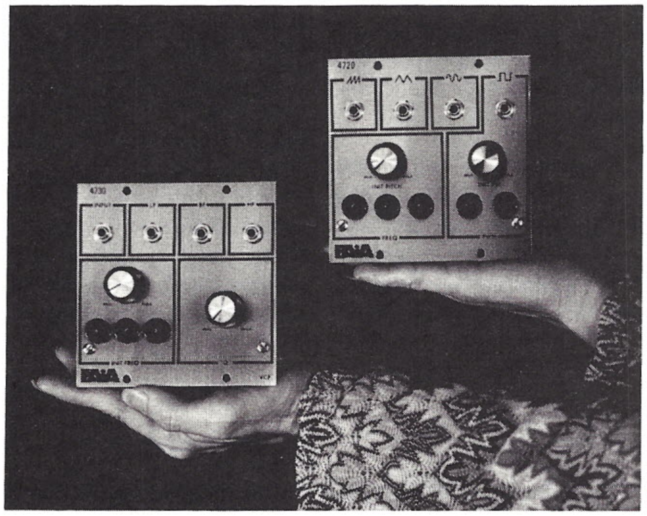


EQUALLY TEMPERED DIGITAL TO ANALOG CONVERTER

Unlike more conventional R-2R ladder type digital to analog converters, the PAIA 8780 kit is based on a multiplying principle that allows the module to generate the exact exponential stair-step function required to make even the simplest linear response oscillators and filters produce equally tempered musical intervals. The 8780 uses only six bits of data to generate over 5 octaves of control voltage. In an 8 bit system, the remaining 2 bits are ordinarily reserved for trigger flags, but may be used to extend the range of the converter or provide micro-tonal tunings.

The module is physically and electrically compatible with the complete line of PAIA music synthesizer modules and is easily interfaced to any micro-processor with or without hand-shaking logic.

#8780 D/A CONVERTER Kit..$34.95 (plus $1.00 postage)



MUSIC SYNTHESIZER MODULES

PAIA offers a complete line of low-cost voltage controlled music synthesizer module kits including the 4720 Oscillator and 4730 filter shown. Both units feature linear freq./control voltage response and 16 Hz. to 16 kHz. range.

The 4720 VCO produces ramp, triangle, sine and pulse waveforms. The companion 4730 VCF is a state variable design with simultaneously available low-pass, band-pass and high-pass outputs, all with "Q" adjustable from .5 to 150.

Other modules in the series include: Voltage Controlled Amplifiers, Balanced Modulators, Envelope Generators, Reverb Units, Noise Sources and Power Supplied.

All modules are compatible with the PAIA 8780 Equally Tempered DAC for easy computer/micro-processor/micro-controller interface.

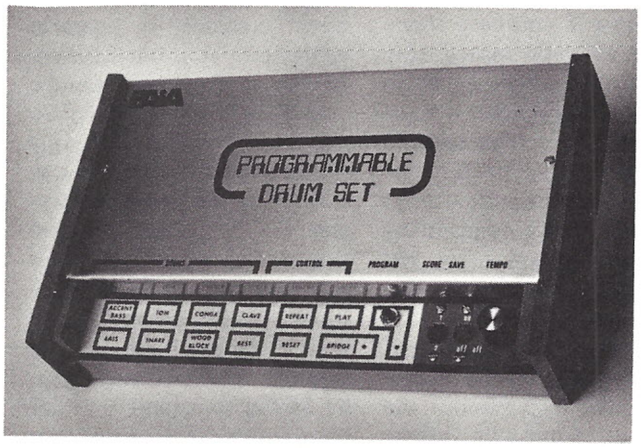
#4720 VCO Kit.......... $34.95 ....(plus $1.00 postage)
#4730 VCF Kit.......... $37.95.......... (plus $1.00 postage)

COMPUTER TECHNOLOGY APPLIED...........
....IN A SELF-CONTAINED PROGRAMMABLE DRUM SET

While most electronic rhythm units offer only a limited choice of pre-determined rhythm patterns, the PAIA Programmable Drum Set allows the user to tailor pattern, time signature and drum sounds to each application. Among the unique features provided by the unit are touch sensitive electronic controls and the provision for an independently structured bridge rhythm.

Battery powered, the drum set includes a "memory save" switch which provides a lowered "keep-alive" voltage to the drum set's 256 byte memory.

#3750 PROGRAMMABLE DRUM SET Kit $79.95
(plus $3.00 shipping & insurance)



All these and more in our FREE CATALOG.......

FROM THE INNOVATORS AT:

PAIA ELECTRONICS
DEPT 4-K
1020 W. WILSHIRE BLVD.
OKLAHOMA CITY, OK 73116

The Programmable Drum Set and all Synthesizer Modules are designed to play through any hi-fi or musical instrument amplifier.

nate speaker. Which brings us to yet another troubleshooting fundamental.

11. One troubleshooting technique that is fundamental is called substitution. If a component is suspected of being faulty, substitute a known good component. I think that I first heard this jewel when I was in the service. How does one get this known, good component? Obviously, if a circuit is functioning, all the components must be good. So all we have to do is take a functioning speaker out of another piece of equipment and stick it in our circuit to test our speaker. Sometimes this is practical, most of the time it is not. However, the principle of substitution is a sound one, and is used extensively by every technician in his troubleshooting "bag of tricks." However, this is the way it usually works: We keep on substituting parts until we get the circuit functioning, and then we put back into the circuit, one at a time, all the different parts to determine which ones are good and which ones are bad. Nuff along those lines for now.

12. OK, the speaker is good, but our baby still won't "play." Can we check the capacitor that couples energy to the speaker? Yes, there are a couple of things that we can do. First, try several other capacitors if you have them. If one of them makes baby play, then we have a way to test other capacitors. If none of them works, then we need to find a way to test the capacitor. A capacitor stores electrons. If we connect the capacitor directly to the power supply terminals, electrons will flow from the power supply into the capacitor and charge up the capacitor. If we then remove the capacitor from the power supply, the stored electrons will be "trapped" inside the capacitor. Connecting the capacitor directly across the terminals of the speaker will allow the capacitor to discharge and the rush of electrons through the speaker voice coil will again cause a click in the speaker. We could have used this trick above to test the speaker, but we were trying to test the speaker by itself and not both the speaker and the capacitor. There are a couple of things wrong here, and this could lead us to problems. If the capacitor isn't large enough (doesn't have enough mfd of capacity), then we can't get enough electrons stored in the capacitor and we will not be able to hear the click. So the best technique available to us at this point in time is to substitute several capacitors. If none of them work, then you can be reasonably certain that the capacitor is not the problem. (Note that you haven't proven the capacitor good or bad, but you have increased the odds in your favor that the capacitor is probably not at fault.)

13. Well, we've got two halves of the remaining circuit left to check: the booster half and the clock generator half. Baby still doesn't play. Since we have three sections in parallel, the booster couldn't be at fault. Whoa, why not? If one section of our IC is shorted, then we still wouldn't get from the clock generator to the speaker. Earlier, we had you take out inverter sections to evaluate how much boost three sections in parallel would give you, but we were talking at that time about a circuit that was operating. So now we connect just one section of the inverters one at a time to the buffer section and see if we get a signal through. If we do, then we can find out which of these three sections of our chip is bad. If we don't get a signal through, then we will again be forced to assume that the booster Amp is not where the problem is. We can do this because we have created odds of 3:1 that this section is good by our troubleshooting techniques. We haven't proven our problem yet, but we have increased our chances that this portion of the circuit is good.

14. Well, we are getting down to the nitty-gritty now. Can we divide what is left into two halves? Yes, we have the clock generator section and the buffer Amp. If the buffer Amp is bad, we can "prove" that section by substituting one of the inverter sections in the booster amplifiers for the buffer. And we can use the same technique for each section of the clock generator. Rewire the inverter sections one at a time utilizing another inverter section.

15. OK, Young, what now? It still doesn't work. Well, when the "Old Man" (ham talk, Old Man or OM is the spouse of the XYL) reaches this point, then I still have one more trick. Tear the whole circuit down and try again.

16. You've all heard the expression, "you can't see the forest because of all the trees." Well, this can happen in electronics too; we work so long and so hard on a circuit that we keep making the same error over and over again and can't see it. So many times, when the circuit doesn't work, I tear it all down and start all over again after a short break.

17. Now we have really reached the end. If our circuit doesn't work still, we have to get some help. At this point I cannot be of much assistance to you unless you want to mail me your entire experiment, parts, batteries and all. This will put you "out of business" while things are in the mail going both ways, and it certainly won't be inexpensive. Try to get some help from some of those resources that I mentioned earlier. A ham will help. The local high school or junior college electronics instructor will help. Even Mom, Dad, wife, husband, friend can sometimes help if you talk the problem over with them, even if they don't know as much electronics as you do.

While this section has been longer than I anticipated, all of the fundamental principles of troubleshooting have been discussed. The concepts of

Bare Budget	**Loaded Budget**
1) Superstrip, SS-2, Nickel Contacts $17.00 (1/3/77)	Superstrip, SS-1, Gold Contacts $18.90 (1/3/77)
2) Soldering Iron, 20-25 W. Mouser has a Marksman, at about $5.00	Weller WTCP-L with PTA6 Tip at about $45.00
3) Screwdriver, Xcelite R3163 at about $1.25	Same
4) Long Nose Pliers, Xcelite 51CG at about $3.25	Same
5) Wire Strippers, Xcelite Model 100 at about $1.75	Same
6) Solder, Electronic Solder is too heavy to mail, pick up locally. I recommend Ersin Multicore, Savbit, 60/40, 18 or 20 gauge. It costs dearly but it is the best that I have used. Sorry about that, American Manufacturers.	
7) Hook-up wire, #22 or #24, assorted colors of insulation. Buy locally. Also see scrounging in text.	
8) Integrated circuits, LEDs, diodes, etc. Usually called the active components. The list below includes those necessary for the two first class sessions. 2 each: 7404, 7400, 7473. 1 each: 7448, FND 70 R.O. silicon diode, ½ Amp, 50 PIV. 6 each: LEDs.	

Table 1. List of Materials for first two class sessions.

dividing the circuit in ½, then in ½ again, and then in ½ again and again and again, until you have things narrowed down to one of two possible items is called *isolation* in the textbooks. The fundamental principle of noncircuit operation because of missing power supply voltages, or incorrect voltages, has been covered. And the concept of working from the output back to the input has also been illustrated. Everything you do should be logical, and you should have a reason for every step that you take.

KILOBAUD KLASSROOM
Experiment #3
The Light Emitting Diode

Purpose:

1. To investigate the Light Emitting Diode (LED).
2. How does one test an LED?
3. How can we test ICs with an LED?

Equipment:

1. Light emitting diode

New Symbols

LED symbol. Refer to Fig. 6a. Notice first that we retain the basic diode symbol. Second, that the basic symbol is enclosed in a circle. And third, note that there are two (sometimes more, sometimes only one) wiggly lines next to the symbol. These wiggly lines are light rays, and they signify that this device will give off light when it is energized.

Circuit

Fig. 6b shows the circuit used to test an LED. The resistor is a current limiting resistor and *must not be omitted*. Never connect any kind of diode directly across the power supply terminals. The current limiting resistor should be around 150 Ohms for a 5 volt power supply. Values as low as 47 Ohms and as high as 470 Ohms are permissable. Connected as shown in Fig. 6b the LED is forward biased and will give off light. Fig. 6c shows the circuit used to test an IC.

Procedure

1. Test the LED. Use circuit shown in Fig. 6b. The LED is proven first.
2. Test the 7404 IC. Leave the - end of the LED connected to - after testing the LED in step 1 above. Connect a 150 Ohm resistor and 6" of wire to the + end. Connect the other end of the resistor to pin 2 of the 7404. The LED should be off. With another piece of wire ground pin 1 of the 7404. The LED should turn on. Repeat this procedure for the remaining 5 sections of the 7404.

Theory

When Transistor-Transistor Logic (TTL or T^2L) Logic circuits are connected to power (pin 14 to + and pin 7 to - , all *input* pins assume a high logic level (which is called a "floating" condition; i.e., no connection). If you measure this level with a voltmeter, it will measure about 1½ volts. Since the input to the inverter is Hi, the output is Lo. By grounding the input pin, the inverter inverts, and the output goes Hi, lighting the LED. This is a functional test of the 7404 and can be performed very quickly.

Let us re-label the input and the output as A and B respectively as we have done in Fig. 6d.

The inverter function can be stated: *The output is not the input.* Or: If A is Hi, B is Lo. If A is Lo, B is Hi. Now a Hi for TTL circuits is any voltage near +5 volts. A Lo is any voltage near ground. A Hi is also called a 1, while a Lo is a 0. These voltages are approximately correct for TTL for positive logic. Another way to express this relationship is to use a mathematical type equation. We could write $A \neq B$ which says that A does not equal B. We can also write $A=\overline{B}$, which also says A is not B. The bar over the B is read as *not*. This last method of stating that the output is not the input is called Boolean Algebra.

These are all different ways of saying the same thing. The small circle following the triangular inverter symbol also is interpreted as *not*. In a subsequent experiment we will introduce you to the *truth table* which is still another way to express logic relationships. What you, the student must do, is to determine which of these methods is easiest for you to comprehend and learn that method thoroughly. You still need to be aware of the other methods so that you can understand the work of others.

One last point: If you find one, or even two sections of an IC bad, don't throw it away. Simply cut off the pins of that IC section which is defective (so that you will not spend time unnecessarily troubleshooting an inoperative circuit) and save it for those circuits that only require part of an IC for operation. If you bought the IC from a reputable dealer, he will of course replace the defective IC without charge. However, 13¢ mailing one way and 13¢ mailing back for a 25¢ IC = $\overline{\text{Logic}}$.

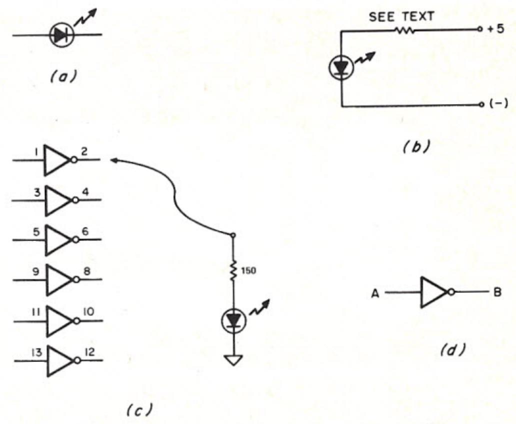


Fig. 6. Circuit diagrams and symbols for Experiment #3.

KILOBAUD KLASSROOM
Experiment #4
Resistance and Capacitance in Series and Parallel

Purpose: To investigate resistors and capacitors in series and parallel.

Equipment: No new equipment is needed.

Symbols: No new symbols are introduced.

Circuit: Refer to Fig. 7a which is almost identical to the circuit for experiment #1.

Procedure

Start by redoing the circuit of experiment #1. Add the LED shown from pin 13 to ground. Energize the circuit. The LED is flashing on and off. It is doing so at the audio rate of the circuit clock. It will appear to be on continuously, but may be slightly dimmer than it was when the IC was being tested. Interchange the positions of the two capacitors in the circuit. That is, put the larger capacitor between pins 1 and 4, and the smaller one between pin 10 and the speaker. The larger capacitor slows down the clock so that you can now see the LED flash. The speaker will now be producing clicks instead of a tone. Now change the value of the resistor connected between pins 1 and 2 on the IC. A value of 270 Ohms and a 100 mfd capacitor produces about 6 Pulses Per Second (6PPS) or 6 cycles per second (6 cps) or 6 Hertz (6Hz). A resistance of 470 Ohms will lower this to about 4 PPS while 150 Ohms will increase this to about 18 PPS. Put the 270 Ohm resistor back in the circuit. Now take another resistor (say the 470) and place it in parallel with the 270 Ohm resistor (it will also connect to pins 1 and 2 when it is in parallel). The number

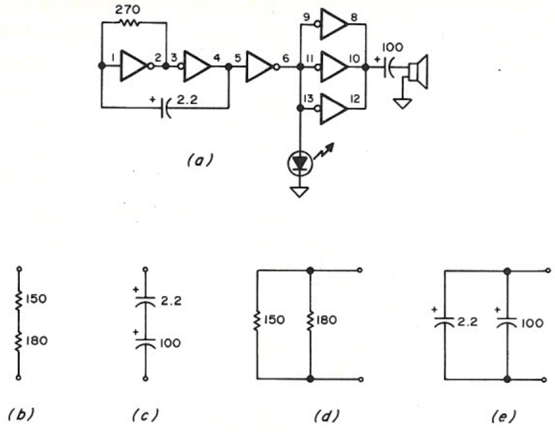


Fig. 7. Circuit diagrams and symbols for Experiment #4.

of PPS will increase. This is because resistors in parallel combine to produce a resistance value that is less than the value of either resistor used alone. Now take any resistor that you have on hand and connect it in series with another resistor. Connect this series combination between pins 1 and 2. This combined value must be less than 470 Ohms or the clock circuit will surely stop oscillating. Next jumper one of these resistors with a piece of wire, effectively taking it out of the circuit. The PPS rate should again increase. Remove the jumper, the PPS rate should decrease. This is because resistors in series add together. A 150 Ohm and a 270 Ohm resistor connected in series will produce a 420 Ohm resistor.

Next take any other capacitor that you have on hand and place this capacitor in parallel with the timing capacitor connected between pins 1 and 4. The number of flashes of the LED should decrease. This is because capacitors connected in parallel add their values together. Connect these two capacitors in series between pins 1 and 4. The PPS rate should increase. This is because capacitors in series combine so that the effective capacitance in the circuit is less than the value of the smallest capacitor alone. Fig. 7b shows resistances in series, they add. Fig. 7e shows capacitors connected in parallel, they add. Figs. 7c and 7d show capacitors in series and resistors in parallel, respectively, they combine reciprocally. The formula for reciprocal combination is:

$$\frac{1}{C_t} = \frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_n}$$

for capacitance and

$$\frac{1}{R_t} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_n}$$

for resistance.

To be perfectly honest about it, only electronic teachers and professors bother with these formulae. For all practical purposes, you need only to memorize the following two statements:

1. Two resistors of the same size connected in parallel, *or* two capacitors of the same size connected in series, produce an equivalent value equal to ½ the value of either the resistor or one capacitor used alone. Example: Two 100 Ohm resistors are connected in parallel. The resultant resistance is 50 Ohms.

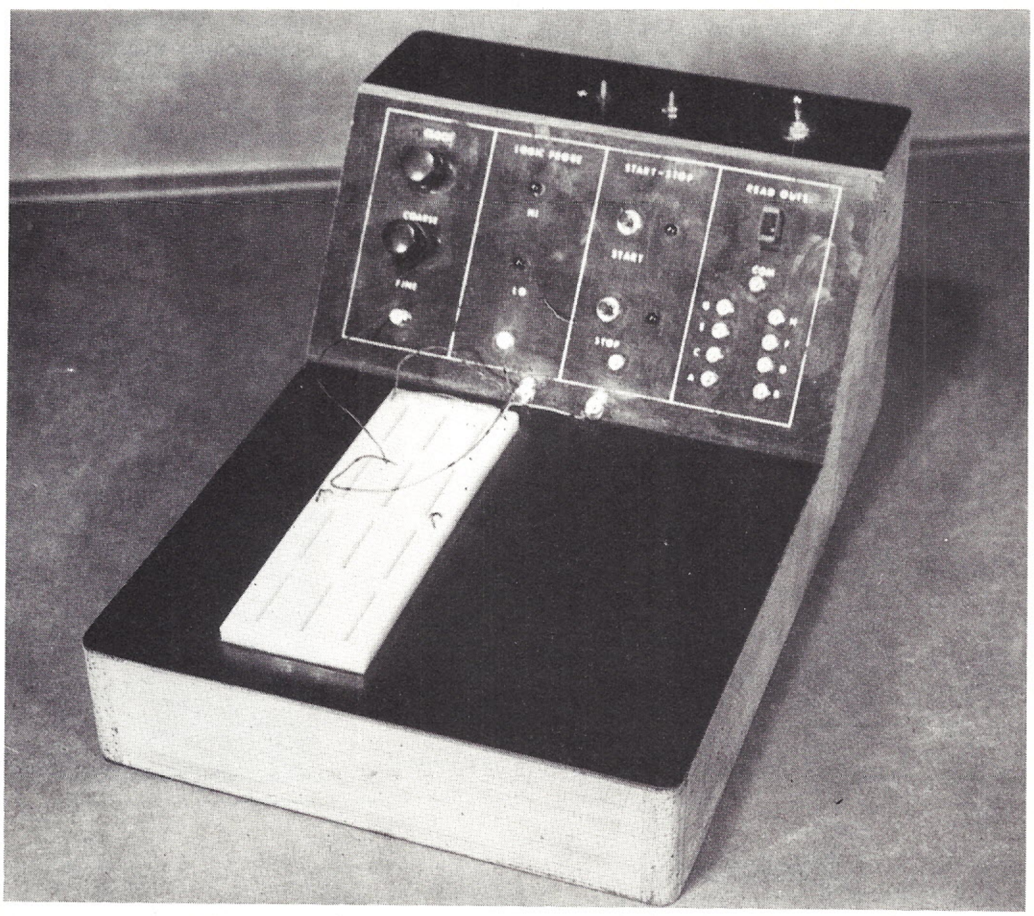
2. Two resistors of unequal value connected in parallel, *or* two capacitors of unequal value connected in series, produce an equivalent value of the resistance or capacitance which is less than the value of the smallest resistor or capacitor used alone.

KILOBAUD KLASSROOM Preview

Coming up next class session: 6 experiments on a) the 7400 gate b) the electronic color code c) "haywiring" of circuits d) a simple logic probe e) the 7-segment readout f) the flip-flop.

We will begin construction of an expanded breadboard. We call this our console. You'll need some sugar pine about 1 x 12 x 12 or so, preferably planed to ½ inch thickness. If sugar pine is not locally available, any soft wood will do. (Surprise – lumber still has not even begun to convert over to the metric system.)

And we will be needing some laminated plastic to "skin" our console with. (For



COMING NEXT MONTH . . . George will have an article which is separate from Kilobaud Klassroom (but an integral part of same) describing the construction of this student console and breadboard station. Needless to say, this unit will serve you well throughout the Kilobaud Klassroom experiments and for many years to come. – John.

$5 I hauled home in the back of my car about 150 lbs of Formica scrap from a local cabinet shop.)

We'll also need a toggle switch, either standard size or subminiature, which can be Single Pole-Single Throw (SPST) or Double Pole-Double Throw (DPDT) or whatever. A couple of dozen 4-40 nuts and bolts about ½ inch long, and two about 1 inch long. A couple of inexpensive push-button switches. And a real toughy – a single pole, 4 position rotary switch. (Don't panic if you can't find this one, I'll show you a way to get around this requirement.) We will also need a volume control (pot) around 200, 000 Ohms. Any value from 50k to 500k can be used. The taper on the pot is not very important. So scrounge up as much as you can while waiting for your next issue of *Kilobaud* to arrive.

Where To Get Parts and Supplies

Try and get the wire and solder locally if you can. We have contacted several suppliers and asked them if they would be agreeable to a 10% discount for you if you mentioned Kilobaud Klassroom with your order. Some said no and some said yes. The yes answers have been tabulated. Mouser Electronics and Dunlop Electronics were contacted as possible tool sources. Mouser did not want to be nailed down on any prices (and I don't blame them) but they will send you their catalog free if you write and request one. Write to James R. Young, Jr., Mouser Electronics, 11511 Woodside Ave., Lakeside, CA 92040, and ask for their latest catalog and prices. Mouser has a $20.00 minimum order but considering the high cost of quality tools these days, you may not have to worry about this minimum. Mr. Young did not mention the 10% discount requested in his letter. You'll have to ask him yourself when you write for Mouser's catalog.

John Toro, of Dunlap Electronics, 1750 E. McKinley, Fresno, CA 93703 offered the following: 1 Weller, Model SP23 25 W. Soldering Iron; 1 Xcelite Model 41 CG Long Nose Pliers; 1 Xcelite Model 55 CG Diagonal Sidecutters; 1 Xcelite Model 100 Wire Strippers; 1 Xcelite Model R 183 Screwdriver; 1 Xcelite Model R 3163 Screwdriver; Package price, including shipping is $19.

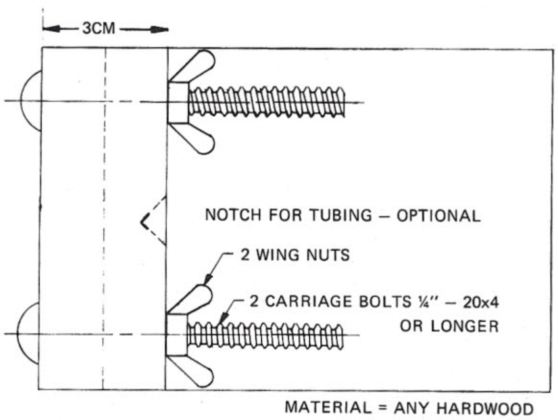
But with the following stipulations: 1) The package is a special package price, minimum order for Kilobaud Klassroom. 2) Price includes the discount mentioned in the text above. 3) Kilobaud Klassroom must be mentioned with your order to insure processing of the order at the special package price. 4) Money order, or certified check must accompany your order. 5) Insurance, shipping or postage will be paid by Dunlap Electronics (within continental U.S. only). 6) Conditions and prices subject to change without notice.

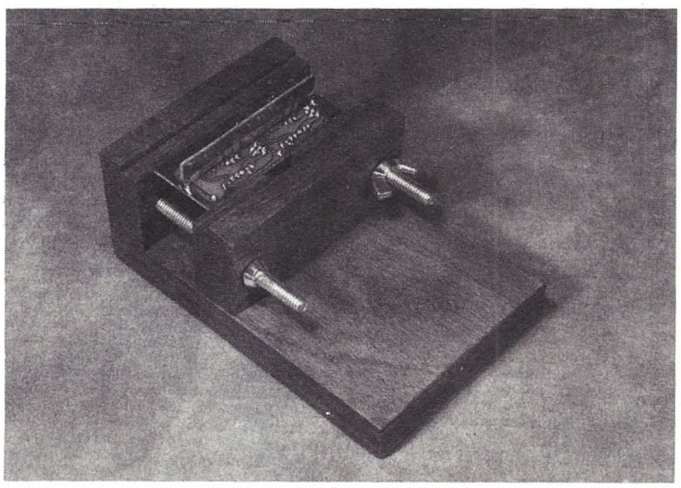
Please note that this tool list has *two* more tools listed in it than we have listed in the Bare Budget list.

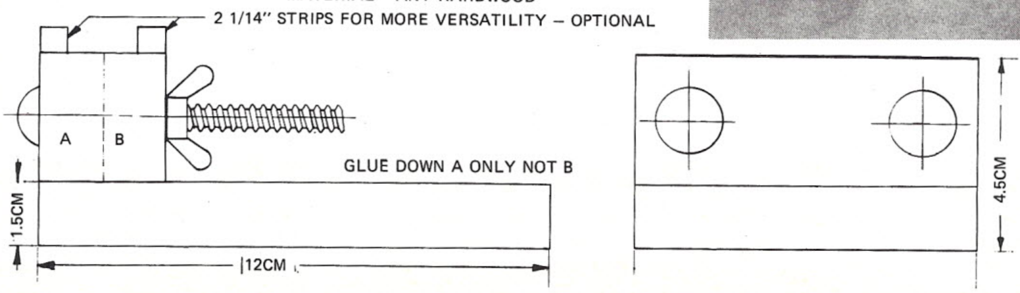
The active components (ICs and LEDs) can be purchased from any of the advertisers in *Kilobaud*. Let me forewarn you that buying the HEP and SKA lines at your local electronic supermarket is not going to be the way to go.

Sierra Electronics, Box 11, Auberry, CA 93602, has agreed to package the active components for us. The package for experiments 1-4 which includes just two components and the Superstrip is $20 for the gold-plated strip. The active component package for the second class session is $5. (Refer to Table 1 for a complete list of tools and components.) California residents must add 6% sales tax. And be sure to allow for postage and insurance with all your orders. Don't fail to mention Kilobaud Klassroom with your orders or they may not get processed at all; your money may be simply returned with a polite note of explanation. For example, Sierra Electronics is a firm that deals in industrial supplying. They have a minimum order of 100 units of each type. While the order for the Superstrip would probably key them in that it was a Kilobaud Klassroom order, it is still advisable to mention the class with every order placed.

And, of course, we need you to scrounge up resistors, capacitors, speakers and anything else you can get your hands on. Get your junk box built up. ■







Electronic bench vice, cost: 35 to 45¢. (Drawing by Scott D. Penner.)

We Just Can't CRAM it ALL in Kilobaud!

Yes, there are computer articles in *73* . . . a lot of them. There are also a lot of articles that computer hobbyists will be needing to read which are not exactly computer articles such as on regulated power supplies . . . on making printed circuit boards . . . on how various circuits work . . . things like that which hardware men in particular need to read . . . and which software people need even more, since they are a bit behind on hardware.

73 is written for the average ham . . . and that means that the level is not PhD by any means. The level of articles in *73* is quite parallel to the level of computer articles in *Kilobaud* . . . and that means that you will be able to understand them and profit from them.

There are computer application articles . . . oriented towards hams, of course. Hams also need to understand the basics of computers, so these are also being covered.

During the last year or so there have been over 300 pages of computer articles and nearly as many which are of interest to the average computerist.

Take the March 1977 issue of *73* just as an example. The big feature was a high quality video display with complete cursor control and video control. This was by Don Alexander, the winner of the WACC exhibition last year. This generates upper and lower case, and even Greek letters! 6800 users will be excited about the operating system described in this issue . . . complete with the hex listing . . . which is used right along with Mikbug and greatly increases the flexibility of the system.

There's an article on using ICs . . . one on a fantastic low voltage power supply with overcurrent protection . . . a capacitor comparator . . . the 79MG and 78MG new breed of voltage regulators . . . a PROM message generator for RTTY . . . how counter ICs works . . . a speedy audio counter . . . making your own PC boards . . . things like that.

In other recent issues there have been articles on computerized satellite tracking (with software), RTTY using a uP, using old (inexpensive) Teletypes, building a Polymorphice video board, making instant PC boards using the new color-key technique, the TTL one-shot, what computers can and can't do, a hamshack file handler (software), the bit explosion — 8-12-16?, backward branch the easy way with the 6800, the hexadecimal . . . etc.

Any one of these articles could easily be worth the cost of a full year of *73*. One good program could save you days of work. One good interface project could make an enormous difference. In general, *73* tries to present not too complicated construction projects . . . things you can make in a day or two.

HAM MAGAZINES

There are a number of ham magazines and they all have one thing in common . . . hardly anything for the computer hobbyist . . . except for *73*. *73* has been running an I/O section since early 1976 . . . computer articles . . . and they are still coming.

One of the fundamental policies is that no articles will be published in both *73* and *Kilobaud*. This is, in a way, unfair because it keeps some great computer articles away from computerists. But since about 20% of the readership of the two magazines overlaps, it would be unfair to those getting both magazines to duplicate. You really must get both magazines to keep up to date with what is going on. When you subscribe to both you will not be getting duplication.

BIGGEST–BEST!

73 VS KILOBAUD

Kilobaud has been outstanding because it is so filled with articles of interest. You've probably noticed that you don't finish *Kilobaud* very quickly . . . and that it takes a lot longer than most other hobby magazines. You'll find the same thing with *73*. Sure, it is ham oriented . . . but remember that ham radio is about 30 different hobbies . . . and today that includes computers.

Look at it this way . . . if you decide you don't want to get *73* you can cancel your subscription and get a refund on the unused parts. You *will* enjoy *73* . . . and you might even get sucked into hamming . . . you could do worse.

SPECIAL FOR KILOBAUD SUBSCRIBERS

The newsstand price is $2 per copy . . . that's $24 a year. The regular subscription rate is $15 for a year. If you are already a subscriber to *Kilobaud* then you are eligible for the special $12 for one year subscription to *73* . . . U.S. and Canada only. This offer is limited and will probably not be repeated once we take a good look at the increased postage and printing bills. Take advantage of us while we are in a weak moment . . . subscribe.

HALF PRICE SPECIAL

☐ YES! Enter my one year subscription to 73 Magazine starting with the next published issue . . . at half the newsstand price . . . just $12.00.

Name ______________ Call (if any) ______________

Address ______________

City ______________ State ______ Zip ______

$12 enclosed. ☐ Cash ☐ Check ☐ Money Order

Bill: ☐ Master Charge ☐ BankAmericard ☐ American Express

Card # ______________ Interbank # (MC) ______________

Signature ______________ Expiration date ______________

☐ Bill me direct

Allow 6 weeks for subscription processing.

Toll free number: 800-258-5473

73 MAGAZINE PETERBOROUGH NH 03458 K/5/77

Glossary

Tim Barry

BAUD: In general communications engineering a baud is the time duration of the shortest element in a code. The baud rate is the number of these code elements that are transmitted in one second. If the signal has only two states (i.e., normal binary signals found in common use with UARTS, 20 mA current loops, and so on), one baud equals one bit. In this case the baud rate equals the number of bits/sec. Thus 30 cps in the standard 10 bit/character ASCII code system = 30 cps * 10 bit/character = 300 baud. In a multiple level code system the bit rate and baud rate will be different. For example, an eight level code will require three bits to represent each unique state ($2^3 = 8$). One baud is now equal to three bits.

BREADBOARD: A special type of circuit card upon which circuit prototypes are constructed. The circuit is debugged on the breadboard because it makes it much easier to make corrections and modifications. Once the design is complete the circuit will often be layed out as a printed circuit card. This comes from the early days of amateur radio when young hams would swipe a breadboard from the kitchen to build an experimental radio circuit. Today any test setup where wires can be moved around or in which a lot of clip leads are used is called a breadboard. Once a circuit is debugged on this test board it can be made into a more permanent printed circuit . . . and probably won't work.

COMPUTER HACKER: A term originally coined back in the 1960s to describe computer professionals. Generally considered obsolete except for a certain small systems journal which continues to use it.

DATA FILE: 1. A collection of data elements stored together in the computer system. The entire file may be used in various operations (file copy, merge, concatenate, delete, etc.) or the individual elements may be operated upon individually (record read, write, alter, etc.). 2. A sharp metal object used to smooth the edges of rough data.

KLUGE: A patched, bandaged, spliced, and otherwise makeshift piece of hardware or software. Kluge performance is not usually spectacularly reliable, and to characterize someone's pet design as a kluge is generally asking for a probe in the back. Kluges often result when a design that worked well in one application is subsequently adapted, expanded, or "improved" to fill another need.

MIKBUG®: MIKBUG is a small debug monitor developed by Motorola Semiconductor for use with their M6800 microprocessor. It resides in a one thousand byte metal masked ROM, and it is provided as part of the resident software for some 6800 based systems. MIKBUG provides only a fairly limited set of operations, but a significant library of 6800 software has been developed which uses some of its internal routines. It also provides the paper tape I/O drivers for use with programs in the M6800 users library. MIKBUG is a registered trademark of Motorola.

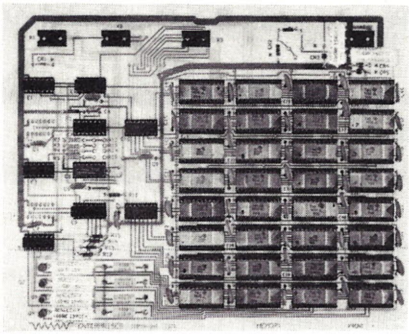
OEM: An Original Equipment Manufacturer (OEM) buys parts and subassemblies from other manufacturers. He then combines them together to form more complex and valuable products which he then sells.

PL/M®: A higher level microcomputer compiler language originally developed by Intel for the 8008 microprocessor. It is a subset (a very *small* subset) of the large system PL/1 language. It features block construction, limited arithmetic/logic functions, and good program control constructs. Intel updated PL/M to support the 8080 when it became available, and other manufacturers have developed languages like PL/M for their microprocessors. These are currently available for the Z-80, M6800, S2650, among others. PL/M is a registered trademark of Intel Corporation.

RATS NEST: A circuit board or lab bench which is covered with a large jumble of wires, cables, and cords. Rats nests are very prone to noise problems, and they can cause very erratic circuit performance (see kluge).

SECOND SOURCE: A second source is a manufacturer who produces equipment that is completely interchangeable with another manufacturer's. The term is most commonly encountered in auto parts and integrated circuits. A good portion of the success of the TTL integrated circuit family is due to the almost universal second sourcing of all parts. This drives cost down and ensures that the demise of one manufacturer will not end production of the part. Any electronic design which uses integrated circuits which are not second sourced (and preferably multiply second sourced) runs the risk of serious supply and/or price problems if the sole source manufacturer is unreliable.

• 270 nsec Access Time • 470 nsec Read/Write Time • TTL Compatible Address Bus • Tri-State Data Bus Driver • Fully Socketed • Sphere Compatible • Easy Home Brew Interface • Voltages +12, +5, -5 •



LOW COST MEMORY 16K x 8 BIT DYNAMIC RAM

Model	Description	Price
WWW-16KA	Fully Assembled	$650.00
WWW-16KK	Kit	$550.00

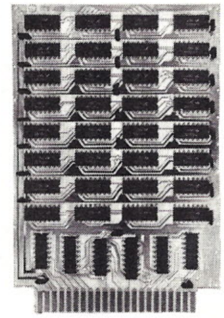
W W W ENTERPRISES
P.O. Box 548,
Harbor City CA 90710
(213) 835-9417

W-14

4K RAM BOARD KIT

FAST, LOW POWER 2102-1 (450 NS)

DENSE 4.5" x 6" PACKAGE



FULLY BUFFERED

STANDARD 44 PIN GOLD PLATED CONNECTOR

$79.95

COMPLETE KIT INCLUDES BOARD, CHIPS, CAPS, & DOCUMENTATION

450 ns	low-power 2102-1	$ 1.60
512x8	bipolar prom	$17.00
16K x 1	250 ns dynamic ram	$54.00
NEW :	64K S-100 dynamic ram board	$395.

ASSEMBLED & TESTED WITH 16K RAM

SEND CHECK OR MONEY ORDER

WASATCH SEMICONDUCTOR PRODUCTS
25 SOUTH 300 EAST, SUITE 215
SALT LAKE CITY, UTAH 84111

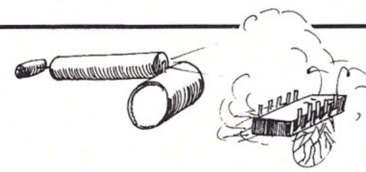
FOR ORDERS UNDER $25.00, ADD $2.00 SHIPPING AND HANDLING

UTAH RESIDENTS ADD 5% SALES TAX

MASTERCHARGE

W13

HELP DEBUG THE WATS LINE



If you've called 800-251-6771 (second number in our ad listings) with an order for merchandise or subscriptions to KILOBAUD and more than six (6) weeks have passed by with you still awaiting delivery or reason for shipment delay notification please send a copy of your cancelled check or credit card bill and itemized list of your order to WATSLINE

c/o kilobaud Peterborough NH 03458

— for immediate handling.

BEST Computer Mailing List

By far the most complete mailing list available is the KILOBAUD list of DEALERS, CLUBS, PUBLICATIONS and MANUFACTURERS. (It's the one we use for our mailings and we update it daily). The list has over 600 names painstakingly gathered from manufacturers, magazine ads and new product releases, hobby computer shows and direct mail. You can buy this list printed on self-sticking labels for only $50.

Additional printouts, once you are a customer, are $35. Call in your order with charge information (BAC, AMEX, MC). Our toll free number for these orders is (800) 258-5473.

NEW FIRMS, DEALERS, CLUBS . . . be sure we have your name, address, phone number and as much data as possible for this listing.

5/77

kilobaud PETERBOROUGH NH 03458

"We may have the surplus electronics you need!"

Terminal/Keyboard
With Schematics
$250.00

- CRT Terminals
- Assemblies
- Peripherals
- Components

Tape Drives $800.00

Keyboards $40.00 to $60.00 (ASCII Encoded)

Equipment Cabinets (19" With Fan) $45.00 to $60.00

Send for a free catalog or call Bill Blaney, toll free 800 258-1036
in NH 603-885-3705

Come to our showroom

VOLUME AND INSTITUTIONAL DISCOUNTS AVAILABLE

WORLDWIDE ELECTRONICS, INC.
10 Flagstone Drive, Hudson, New Hampshire 03051

Send my free catalog to

NAME: ____________________

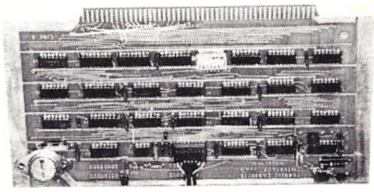
ADDRESS: ____________________

I'M especially interested in:

- ☐ NEW
- ☐ USED
- ☐ AS IS
- ☐ TERMINALS
- ☐ PERIPHERALS
- ☐ COMPONENTS
- ☐ ASSEMBLIES

W16

The Tarbell Cassette Interface



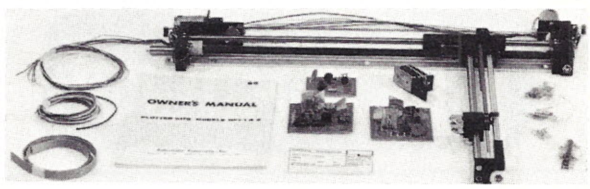
- Plugs directly into your IMSAI or ALTAIR* Computer
- Fastest transfer rate: 187 (standard) to 1000 bytes/second
- Extremely Reliable — Phase encoded (self-clocking)
- 4 Extra Status Lines, 4 Extra Control Lines
- 37-page manual included
- Device Code Selectable by DIP-switch
- Capable of Generating Kansas City tapes also
- No modification required on audio cassette recorder
- Complete kit $120, Assembled $175, Manual $4

*ALTAIR is a trademark/tradename of MITS, Inc.

TARBELL ELECTRONICS
20620 S. Leapwood Avenue, Suite P, Carson, California 90746
(213) 538-4251
California residents please add 6% sales tax

T11

COMPUTER CONTROLLED PLOTTERS!!



90% ASSEMBLED

11 x 17 inch size (Delivery stock) ____________ $750
17 x 22 inch size (Delivery 3-8 weeks) ____________ $895
Owner's Manual (refunded with plotter purchase) ____ $5

- We supply plotter and interface -
- You supply computer and software -
- Dealer inquiries invited -

WRITE TODAY!!

Sylvanhills Laboratory, Inc.
#1 Sylvanway, Box 239 Strafford, Mo. 65757
417-736-2664

S28

GOOD BOOKS?

If you run across a book which you think other hobbyists would profit knowing about, why not write a brief review . . . including the name and address of the publisher (if you have it) and the price and send it to

kilobaud **Peterborough NH**

International Data Systems, Inc.

400 North Washington Street, Suite 200
Falls Church, Virginia 22046 USA
Telephone (703) 536-7373

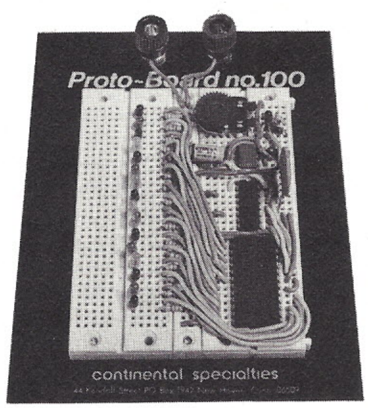
S100 Bus Cards (ALTAIR/IMSAI Compatible)		Uses	Kit Price
88-SPM	Clock Module	Your computer keeps time of day regardless of what program it is executing. Applications include event logging, data entry, ham radio, etc.	$ 96.00
88-UFC	Frequency Counter Module	Measure frequencies up to 600 MHz. Computer can monitor multiple frequencies such as transmit and receive frequency.	$149.00
88-MODEM	Originate/Answer MODEM	Use your computer to call other computer systems such as large timesharing systems. Also allows other computer terminals to "dial-up" your computer.	$199.00
GENERAL PURPOSE PERIPHERALS			
MCTK	Morse Code Trainer/keyer	Hardware/Software package which allows your computer to teach Morse code, key your transmitter, and send prestored messages.	$ 29.00
TSM	Temperature Sensing Module	Use it to measure inside and/or outside temperature for computerized climate control systems, etc.	$ 24.00
DAC8	Eight Bit Digital to Analog Converter	Requires one eight bit output port. Use it to produce computer music.	$ 19.00

Terms: Payment with order. Shipment prepaid.
Delivery is stock to 30 days. Write or call for detailed product brochures. **I-10**

WANT TO HOOK UP A CIRCUIT QUICK?

The PB-100 is only one of our family of solderless Proto-Board breadboarding units, designed to help you assemble, test and modify circuits as fast as you can push in or pull out a lead. *Preassembled* sockets with durable 5-point terminals provide low-resistance interconnections you can arrange and re-arrange at will.

Resistors, capacitors, transistors, DIP's, TO-5's, LED's etc. plug in without damage to leads. And jumper connections, where required, are lengths of #22 AWG solid wire. Models from 630 to 3060 tie-point (6 to 32 14-pin DIP) capacity available. For more information, see your CSC dealer, or write for our catalog and distributor list.



Proto-Board® **100.** 760 solderless tie-points. **Kit, $19.95***

CONTINENTAL SPECIALTIES CORPORATION

CSC

EASY DOES IT
44 Kendall Street, Box 1942
New Haven, CT 06509 • 203-624-3103 TWX: 710-465-1227
West Coast office: Box 7809, San Francisco, CA
94119 • 415-421-8872 TWX: 910-372-7992

© 1976, CSC *Mfrs. sugg. list

TCF-77

THE 2ND TRENTON COMPUTER FESTIVAL
APRIL 30-MAY 1
STARTS AT 10 AM
Trenton State College
Rt 31, Trenton NJ

- **90 Exhibitor Booths**
- **30 Talks/Forums**
- **Huge Outdoor Flea Market**
- **Prizes**
- **Awards for Hobby Exhibits**
- **Awards Banquet — Sat. Nite**

* **Free Software Copy Service**
* **Free Parking for 4,000 cars**

Registration $4 (students $2)
Flea Market Space $2/table

Sponsored by:

Amateur Computer Group of N.J.
Phila. Area Computer Society
Trenton State Digital Society
IEEE — Princeton Chapter
D.I.E.T. — Trenton State College

For info & reservations write or call:
Jaci DiPalo
Trenton State College
Trenton NJ 08625
609-771-2487

Bearcat® 210 $289.

The **Bearcat® 210** is a sophisticated scanning instrument with the ease of operation and frequency versatility you've dreamed of. Imagine, selecting from any of the public service bands and from all local frequencies by simply pushing a few buttons. No longer are you limited by crystals to a given band and set of frequencies. It's all made possible by *Bearcat* spaceage solid state circuitry. You can forget crystals forever.

Pick the 10 frequencies you want to scan and punch them in on the keyboard. It's incredibly easy. The large decimal display reads out each frequency you've selected. When you want to change frequencies, just enter the new ones.

Automatic search lets you scan any given range of frequencies of your choice within a band. Push-button lockout permits you to selectively skip frequencies not of current interest. The decimal display with its exclusive "rolling zeros" tells you which channels you're monitoring. When the *Bearcat* 210 locks in on an active frequency the decimal display shows the channel and frequency being monitored.

With the patented track-tuning system, the Bearcat 210 automatically aligns itself so that circuits are always "peaked" for any broadcast. Most competitive models peak only at the center of each band, missing the frequencies at the extreme ends of the band.

The Bearcat 210's electronically switched antenna eliminates the need for the long low band antenna. And a quartz crystal filter rejects adjacent stations as well as noise interference.

BankAmericard and Mastercharge cardholders can order on our 24-hour toll-free credit card order line 800-521-4414. In Michigan and outside the U.S. call 313-994-4441. Add $5 for U.S. shipping or $8 for air UPS to west coast. Charge cards or money orders only please. Foreign orders invited. Michigan residents add tax. For quantity pricing or if you have any questions, write: Communications Electronics, Engineering Division, P.O.Box 1002, Ann Arbor, Michigan 48106.

Bearcat® 210 Features

- **Crystal-less**—Without ever buying a crystal you can select from all local frequencies by simply pushing a few buttons.
- **Decimal Display**—See frequency and channel number—no guessing who's on the air.
- **5-Band Coverage**—Includes Low, High, UHF and UHF "T" public service bands, the 2-meter amateur (Ham) band, plus other UHF frequencies.
- **Deluxe Keyboard**—Makes frequency selection as easy as using a push-button phone. Lets you enter and change frequencies easily . . . try everything there is to hear.
- **Patented Track Tuning**—Receive frequencies across the full band without adjustment. Circuitry is automatically aligned to each frequency monitored.
- **Automatic Search**—Seek and find new, exciting frequencies.
- **Selective Scan Delay**—Adds a two second delay to prevent missing transmissions when "calls" and "answers" are on the same frequency.
- **Automatic Lock-Out**—Locks out channels and "skips" frequencies not of current interest.
- **Simple Programming**—Simply punch in on the keyboard the frequency you wish to monitor.
- **Space Age Circuitry**—Custom integrated circuits . . . a *Bearcat* tradition.
- **UL Listed/FCC Certified**—Assures quality design and manufacture.
- **Rolling Zeros**—This *Bearcat* exclusive tells you which channels your scanner is monitoring.
- **Tone By-Pass**—Scanning is not interrupted by mobile telephone tone signal.
- **Manual Scan Control**—Scan all 10 channels at your own pace.
- **3-Inch Speaker**—Front mounted speaker for more sound with less distortion.
- **Squelch**—Allows user to effectively block out unwanted noise.
- **AC/DC**—Operates at home or in the car.

Bearcat® 210 Specifications

Frequency Reception Range

Low Band	32—50 MHz
"Ham" Band	146—148 MHz
High Band	148—174 MHz
UHF Band	450—470 MHz
"T" Band	470—512 MHz

*Also receives UHF from 416—450 MHz

Size
10⅝" W x 3" H x 7⅝" D

Weight
4 lbs. 8 oz.

Power Requirements
117V ac, 11W; 13.8 Vdc, 6W

Audio Output
2W rms

Antenna
Telescoping (supplied)

Sensitivity
0.6μv for 12 dB SINAD on L & H bands
U bands slightly less

Selectivity
Better than −60 dB @ ± 25 KHz

Scan Rate
20 channels per second

Connectors
External antenna and speaker; AC & DC power

Accessories
Mounting bracket and hardware
DC cord

™ COMMUNICATIONS ELECTRONICS

Box 1002
Ann Arbor, Michigan 48106 USA

master charge THE INTERBANK CARD

Your BANKAMERICARD welcome here

C5

DIODES/ZENERS

1N914	100v	10mA	.05
1N4004	400v	1A	.08
1N4005	600v	1A	.08
1N4007	1000v	1A	.15
1N4148	75v	10mA	.03
1N753A	6.2v	z	.25
1N758A	10v	z	.25
1N759A	12v	z	.25
1N4733	5.1v	z	.25
1N5243	13v	z	.25
1N5244B	14v	z	.25
1N5245B	15v	z	.25

SOCKETS/BRIDGES

8-pin	pcb	.25	ww	.45
14-pin	pcb	.25	ww	.40
16-pin	pcb	.25	ww	.40
18-pin	pcb	.25	ww	.75
22-pin	pcb	.45		
24-pin	pcb	.35	ww	1.25
28-pin	pcb	.35	ww	1.45
40-pin	pcb	.50	ww	1.95

Molex pins .01 To-3 Sockets .25

2 Amp Bridge	100-prv	1.20
25 Amp Bridge	200-prv	2.50

TRANSISTORS, LEDS, etc.

2N2222	NPN			.10
2N2907	PNP			.15
2N3740	PNP	1A	60v	.25
2N3906	PNP			.10
2N3055	NPN	15A	60v	.50
T1P125	PNP	Darlington		.35

LED Green, Red, Clear		.15
D.L.747	7 seg 5/8" high com-anode	1.95
XAN72	7 seg com-anode	1.50
FND 359	Red 7 seg com-cathode	1.25

C MOS

4000	.20
4001	.20
4002	.25
4004	4.95
4006	1.20
4007	.40
4008	1.20
4009	.25
4010	.45
4011	.20
4012	.20
4013	.40
4014	1.10
4015	.95
4016	.35
4017	1.10
4018	1.10
4019	.70
4020	.85
4021	1.35
4022	1.15
4023	.25
4024	.75
4025	.35
4026	1.95
4027	.50
4028	.95
4030	.45
4033	1.95
4034	2.45
4035	1.25
4040	1.35
4041	.69
4042	.95
4043	1.25
4044	.95
4046	1.50
4049	.80
4050	.70
4066	1.35
4069	.40
4071	.35
4082	.45

— T T L —

7400	.15
7401	.15
7402	.20
7403	.25
7404	.15
7405	.25
7406	.35
7407	.55
7408	.25
7409	.15
7410	.15
7411	.25
7412	.30
7413	.65
7414	1.10
7416	.25
7417	.50
7420	.15
7426	.40
7427	.45
7430	.15
7432	.45
7437	.45
7438	.35
7440	.25
7441	1.15
7442	.65
7443	.95
7444	.55
7445	.95
7446	.95
7447	.95
7448	1.20
7450	.25
7451	.25
7453	.25
7454	.25
7460	.40
7470	.45
7472	.45
7473	.35
7474	.40

7475	.45
7476	.20
7480	.65
7481	.99
7483	1.00
7485	1.05
7486	.40
7489	2.50
7490	.55
7491	1.15
7492	.95
7493	.45
7494	1.25
7495	.85
7496	.95
74100	1.85
74107	.45
74121	.40
74122	.55
74123	.55
74125	.45
74132	1.35
74141	1.30
74150	1.00
74151	.95
74153	.95
74154	.75
74156	1.15
74157	.75
74161	1.25
74163	1.25
74164	.95
74165	1.50
74166	1.35
74175	.95
74176	1.25
74180	.85
74181	3.25
74182	.95
74190	1.75
74192	1.65

74193	.85
74194	1.45
74195	.95
74196	1.50
74197	1.25
74198	2.35
74367	.85
75108A	.35
75110	.35
75491	.50
75492	.50
74H00	.25
74H01	.25
74H04	.25
74H05	.25
74H11	.25
74H15	.30
74H20	.30
74H22	.40
74H30	.25
74H40	.25
74H51	.25
74H52	.15
74H53J	.25
74H55	.25
74H72	.55
74H101	.75
74H103	.75
74H106	.95
74L00	.35
74L02	.35
74L03	.30
74L10	.35
74L30	.45
74L47	1.95
74L55	.65
74L72	.45
74L75	.55
74S00	.55
74S02	.55
74S03	.50

74S04	.45
74S05	.45
74S08	.45
74S10	.45
74S11	.45
74S20	.50
74S40	.30
74S50	.35
74S51	.45
74S64	.30
74S74	.50
74S112	1.50
74S133	.45
74S140	.75
74S151A	.45
74S153	.45
74S158	.45
74S194	1.50
74S257 (8123)	.25
74LS00	.45
74LS01	.45
74LS02	.45
74LS04	.55
74LS08	.45
74LS09	.45
74LS10	.45
74LS11	.45
74LS20	.50
74LS21	.25
74LS22	.25
74LS32	.55
74LS37	.40
74LS40	.55
74LS42	1.75
74LS74	.95
74LS90	1.30
74LS93	1.00
74LS107	.95
74LS153	1.20
74LS157	.85
74LS164	1.90
74LS367	.85
74LS368	.70

9000 SERIES

9301	1.00
9309	.45
9322	1.10
9602	1.50

MEMORY, CLOCKS

74S188 (8223)	3.00
MM1702A	9.95
MM5314	3.50
MM5316	3.95
2102-1	1.75
TR 1602A	6.95
TMS6011NC	6.95
8080AD	19.50
8T23	1.50
8T24	2.00
2107B-4	5.95

LINEARS, REGULATORS, etc.

MCT2	.95
8038	3.95
LM201AH	.75
LM301AH	.25
LM308AH	1.00
LM309H	.65
LM309K	.90
LM310	1.15
LM311D (Mini)	.95

LM320K5	1.65
LM320K12	1.65
LM320T12	1.65
LM320T15	1.65
LM339	.95
7805 (340T-5)	1.00
LM340T-12	1.25
LM340T-15	1.25
LM340T-18	1.65

LM340T-24	1.25
LM340K-12	2.15
LM340K-15	1.65
LM340K-18	1.65
LM340K-24	1.25
LM373	2.95
LM380	.95
LM709 (8, 14 PIN)	.30
LM711	.45

LM723	.45
LM725	1.95
LM739	1.50
LM741 8-14	.25
LM747	1.10
LM1307	1.25
LM1458	.95
LM3900	.65
LM75451	.65
NE555	.50
NE556	1.10
NE565	.95
NE566	1.75
NE567	1.35
SN72720	.35
SN72820	.35

INTEGRATED CIRCUITS UNLIMITED

7889 Clairemont Mesa Blvd. • San Diego, CA 92111 • (714) 278-4394

All orders shipped prepaid — No minimum
Open accounts invited — COD orders accepted
Discounts available at OEM Quantities
California Residents add 6% Sales Tax

24 Hour Phone (714) 278-4394 — MasterCharge / BankAmericard

Hufco PRESENTS THE...

MARK II

FREQUENCY COUNTERS

HOOKUP IS A PIECE O' CAKE with better than 50mv sensitivity, direct connection to the circuit under test is unnecessary in most cases.

FREQUENCIES JUMP OUT AT YOU from the giant ½" readouts.

GREATER FREQUENCY RANGE the 60 mHz typical frequency response gives you 80-10 meters plus 6 meters — 50mHz guaranteed.

AND . . .
YOU'LL FIND ENDLESS NEW APPLICATIONS FOR THE "BURNOUT PROOF" MARK II

With the overload protected front end you can use this counter anywhere in a circuit without fear of burnout. Use the Mark II to test: receiver local oscillators, grid dip meters, RF signal generators, audio generators, touch tone pads (when extend installed), micro-processor timing signals, modems, function generators . . . YOU NAME IT!!!

HUFCO QUALITY AS ALWAYS • SAME HI-QUALITY G-10 GLASS EPOXY DOUBLE-SIDED PC BOARDS • SAME TTL IC'S • MORE THAN EVER . . . AMERICA'S BEST BUY IN DIGITAL FREQUENCY COUNTERS!

The TWS MARK II is available in three frequency ranges:

0-50 mHz - 0-250 mHz - 0-500 mHz

RUSH THIS COUPON TODAY!

This is what I've been looking for: A Goof-proof low cost Frequency Counter! Send me:

☐ **500 mHz Frequency Counter** - 169.95 kit/199.95 assembled
☐ **250 mHz Frequency Counter** - 129.95 kit/159.95 assembled
☐ **50 mHz Frequency Counter** - 79.95 kit/99.95 assembled
☐ **Information on other handy Hufco Products.**

Enclosed is Check - Money Order - BAC/MC Bankcard OK!

Card No. ______________________ Exp. Date __/__/__

Name ______________________

Address ______________________

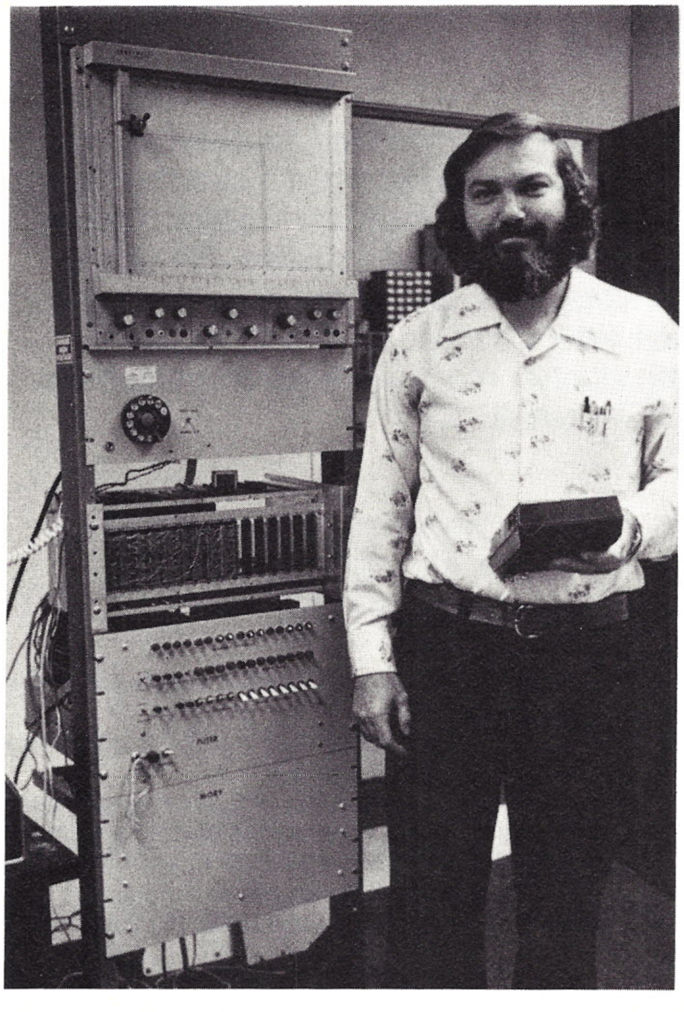
City/State/Zip ______________________

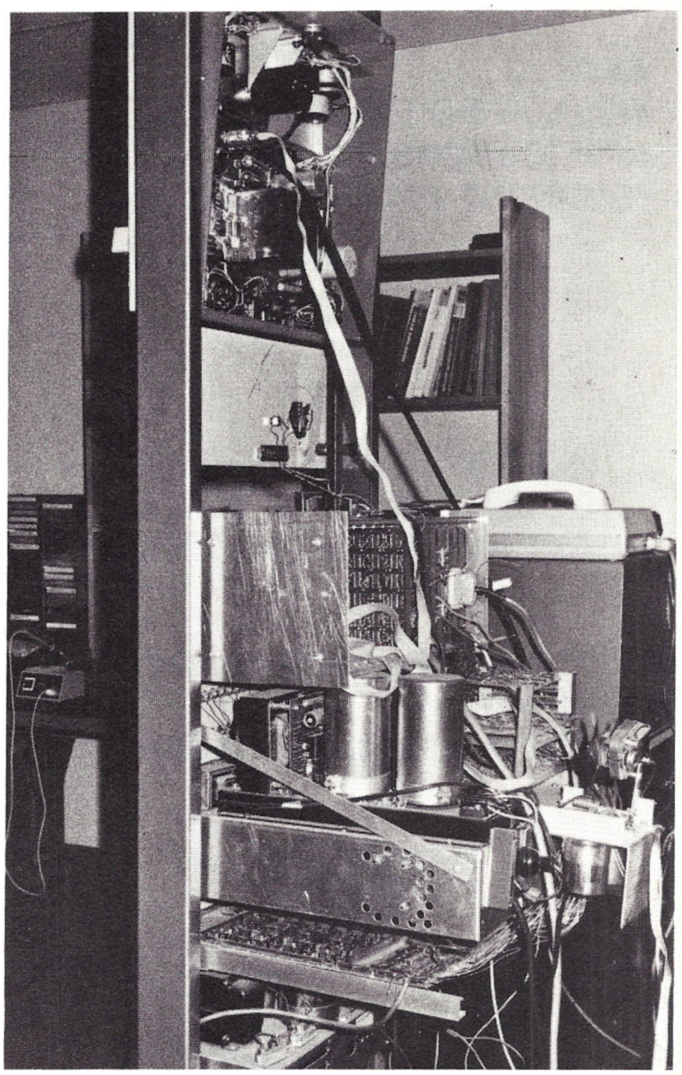
Mail to: Box 375, Dept. K
1603 W. 800 N.
Provo, Utah 84601
801/375-8566

H-10

Sheila Clarke
CYBER GRAFIX
P.O. Box 430
Glendale CA 91206

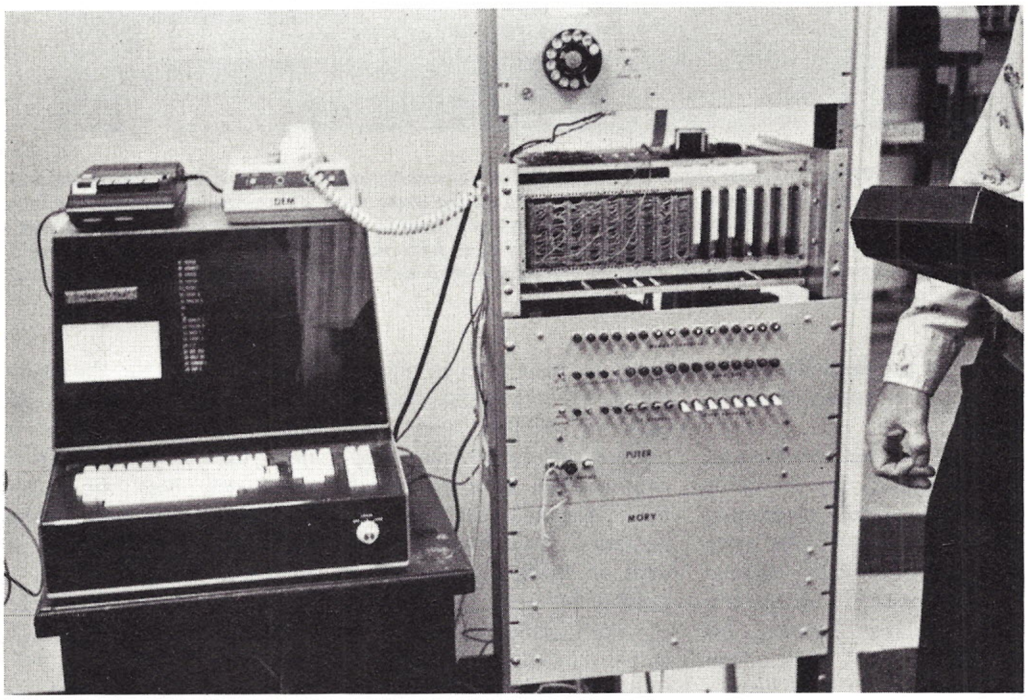
A Home Computer Pioneer





Don Tarbell stands proudly with his home brew system. Built prior to the availability of the microprocessor, it consists mainly of TTL ICs and has interfaces to a CRT, modem, cassette recorder, hard disk, TTY and a plotter. Don modestly permitted a photo of the rear view which displays his talents in organization and assembly.

... profile of Don Tarbell



The CRT terminal on the left was donated to Don by SKI DATA, a result of one of his 1,000 letters. Don's cost: $5 for shipping and the time it took to get it running. Don figures it to be about ten to fifteen years old and says it has an acoustic delay line rather than semiconductor memory.

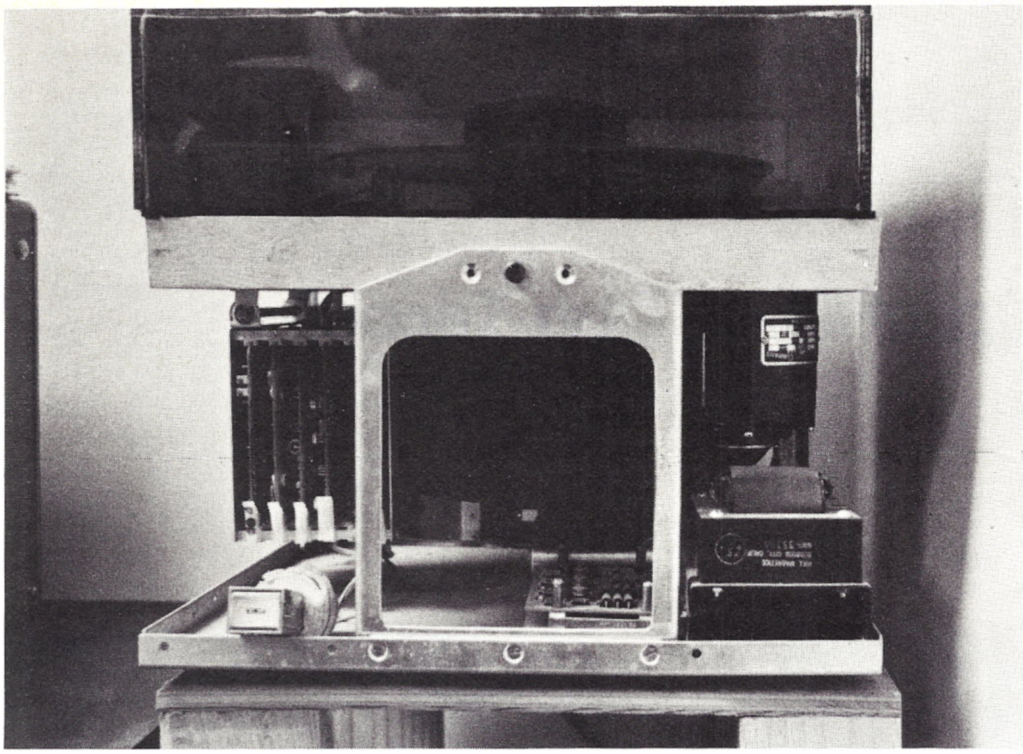
Most folks involved with the microcomputer explosion believe the home computing fever to be no more than two ... maybe three years old. For many, that's about how far back our awareness of small systems goes. Others, however, had foresight and imagination many years ago which filled their heads with notions of owning small systems for personal use. Don Tarbell is one of those pioneers.

No doubt there are few who are not aware of the Tarbell Cassette Interface. Some may even recall that, in the beginning, this device was about the only one of its kind that permitted data storage using so inexpensive a peripheral as a portable cassette recorder. Don was among those computer enthusiasts who were prepared to jump right in at the first announcement of computer kits. He was also one of the people responsible for the forming of the Southern California Computer Society in 1975, now something over 10,000 strong.

People who know Don find him to be pleasant and unassuming, helpful with problems presented him, and with a sense of humor. Because of his unassuming nature, one would little suspect the degree to which he has involved himself in computing; long before most thought personal systems were possible, Don was applying his imagination to building his own.

Back in 1965 when he was fresh out of the Air Force and working for Sperry Rand Corporation, Don decided to return to school at night to complete his work for a degree in electrical engineering. He enrolled at the University of Alabama at Huntsville. Having been interested in artificial intelligence, he figured computers were the only way to handle the task and took a class in FORTRAN.

The programming bug bit hard. Don took one class after another until he'd completed courses in operating systems, machine languages, assembly languages and compiler design. "I thought that was neat," he said. "At the same time I was taking those classes, I started building a little programmable calculator." It was a five digit calculator which could be programmed from paper tape using add, subtract, multiply and divide. He used an old five-level paper tape machine and reader which slowly chunked out the data. Don claims it was "just a clanky machine made with RTL ICs and didn't do all that much. But it could do factorials and would print out on an old Baudot Teletype printer." When asked if he still had the calculator, he said, "No, at the time I was a little hard up for money and sold it for



DATA DISK DRIVE took Don about six months to get it working, but he now uses it daily. With about 75 files on it, it takes only a few seconds to be up and running under complete computer control. Just to be safe, all data is backed up on cassette tape. The disk will hold up to ½ million bytes. It operates with a contact head rather than a flying head and is made of steel with nickel cobalt coating and weighs about 50 pounds. This device is now obsolete, but Don has seen surplus units advertised for $500-$1,000. When new, the disk alone was valued at $800.

$25." He probably had twice that much wrapped up in ICs alone.

While he was working at Sperry Rand as a senior engineer, Don discovered there was more to the game than programmable calculators. He wanted to build a new system. Putting his imagination to work, Don conceived an effective way to obtain the equipment and parts he needed to build his next computer. It occurred to him that many companies must have obsolete equipment they might want to be relieved of. So he sent out about a thousand letters soliciting donations to his project. Don muses, "It's unbelievable that these things were just sitting in the corner collecting dust. Some of these devices are no longer around. I got a CRT display, disk drive, core memory . . . all sorts of neat things through those letters."

One response to his letter came from someone at IBM. The fellow didn't have anything to offer him in equipment, but his advice helped Don in another very important respect. He suggested Don contact a group called the Amateur Computer Society in New York. By this time, in 1966, the group had roughly 50 members throughout the U.S. So Don was now able to multiply his production efforts with the help of ACS members who shared his unique interest.

Don began to design his first real computer, which he called his "microcomputer." It was to consist almost entirely of TTL ICs. "Essentially the idea was to come in below others with minicomputers at a couple thousand dollars, doing the same things on a smaller scale." At this point Don left Sperry Rand to join two others in business for themselves. Some of the new company's work included producing IC breadboards, layout of printed circuit boards for ITT, designing Army digital trainers, and developing a prototype of an analog trainer for the Navy. With 25 employees they felt fairly successful. Then in 1967 the company began to experience cash flow difficulties. Someone came along who was interested in acquiring the small firm. It turned out that the real interest in the company was Don's computer.

So the three sold out and continued on as employees. The business was moved to Nashville where the new company purchased a PDP-8 and a Teletype . . . for over $10,000. Don continued to build his own computer while he did the programming on the PDP-8. In 1970 the entire business was relocated in Miami. The PDP-8, Don's computer . . . everything went, and Don stayed behind. They never finished the project however, having been cut short by the Miami move.

Discovering that mid-1970 was a lousy time to be out of a job (recall the government contract cutback for the aerospace industry?), Don felt this was a good time to go back to school and complete the six hours he needed for his BS degree . . . and do it again! Yup! He wanted to build another computer.

1971 was a good year. Don did get his degree (a BS in Electrical Engineering). He regained his job at Sperry Rand . . . and he got his computer up and running. It was interfaced to a printer, a disk and 4K of memory which could be addressed directly or indirectly. He wrote a disk operating system for it and a test editor which is in use today mainly for printing and updating the pages of his operating manual for the cassette interface.

Back to 1971 . . . Sperry Rand also had a PDP-8. Don's job didn't include using it, but as he says, "I was attracted to it and started playing with it. I got so involved with it, as a matter of fact, that projects on my own job started falling behind." He recalled one incident when he'd wanted to watch a demonstration on a VDM-type display using the PDP-8. Not realizing he'd left a test tape recording device in a cooling chamber, he hurried off to see the demonstration. He recalls, "I got so enthralled with the computer demo that I forgot the test and the temperature of the cool chamber dropped to minus 60°. It was pretty bad . . ."

While Don was learning all he could on the PDP-8 he continued to gather parts and add peripherals to his home brew system.

In 1972 he decided to try his luck in Southern California and applied at Hughes for a job. Before hiring him, however, the prospective empoyer wanted to see Don's computer for himself. The man flew back to Nashville for a firsthand demonstration. Remember, this was in 1972 prior to microprocessors. So Don's computer was quite an innovation and the Hughes official was properly impressed. The man reviewed

COMPUTER WAREHOUSE STORE

DEPT: K • 584 COMMONWEALTH AVENUE • BOSTON, MA • 02215 • 617-261-2701 • VISIT US: 9-9 WEEKDAYS; 9-6 SATURDAYS

ONE DAY SHIPMENT

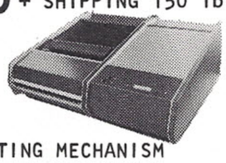
PERIPHERALS FOR MICROSYSTEMS

ALL an ASR 33 is and MORE!

RS232 INTERFACE, QUIET OPERATION, 10 CPS PRINTER, BUILT-IN PAPER TAPE READER/PUNCH, ELECTRIC TYPEWRITER KEYBOARD WITH ADDITIONAL 10 KEY NUMERIC PAD, YOUR CHOICE OF FRICTION OR SPROCKET FEED, LIGHTED PLATEN AREA FOR EASY READING, STANDARD PAPER AND TAPE, SUPPORTED BY OLIVETTI.

$950 + SHIPPING 165 lb

TALLY T132...$450 + SHIPPING 150 lb



7 x 8 DOT MATRIX IMPACT PRINTER HAS A SINGLE LINE DYNAMIC MEMORY AND A UNIVERSAL INTERFACE TO ACCEPT PARALLEL DATA, FORMS TO 14-7/8 IN. WIDE, SIMPLE PRINTING MECHANISM USES 132 SOLENOID HAMMERS AND TWO STEPPER MOTORS FOR 100 LPM, 132 COLUMNS, 64 CHARACTERS

DIGITRONICS D507.....$95 + 400 lb SHIPPING

PAPER TAPE READER, HEAVY DUTY PS, 3 MUFFIN FANS, POWER CONTROL PANEL, CIRCUIT BOARDS, RELAYS IN CABINET

KDI ADTROL AR-21...$95 + $25 SHIPPING

ELECTRO OPTICAL PAPER TAPE READER WITH 110V PS, STEPPER MOTOR, 250 CHAR/SEC, FAN IN TABLE TOP HOUSING

IBM 731 I/O WRITER $750 + $25 SHIPPING

8½"PLATEN,PINFD,EBCDIC,U/L CASE,DUAL CLR RIBBON, 115V

KITS ★ HIGHLIGHTS FROM OUR WIDE SELECTION ★

IMSAI 8080 MICROKITS

8080A KIT 5 SLOT................$699
8080A KIT 22 SLOT............... 751
4K MEMORY KIT................... 139
PIC-8 PRIORITY INTERRUPT........ 125
SERIAL I/O KIT.................. 125
PROM 4-512 KIT.................. 165
UCRI-1 KIT...................... 59
CABLE A KIT..................... 18
STANDARD INPUT/OUT INTERFACES, MANUALS, SOCKET SETS

VIKING 100 PIN, HEAVY DUTY.....$3.00

SWTPC 6800..........$395

- 512 BYTES of ROM
- RS232 or 20 mA
- SERIAL INTERFACE
- 4K of RAM

4K MEMORY.......................$100
MPA...........145 MPB.......... 40
MPC........... 40 MPD.......... 35
MPE.........14.95 MPF.......... 30
MPM........... 65 MPMx......... 35
MPP........... 35 MPL.......... 35
MPS........... 35 MPAb........14.50
MPMb........14.50
MPCb, MPSb, MPLb, EACH..........9.50
CONNECTOR SETS MPU/MEM..........2.50
CONNECTOR SETS INTERFACE........ 2
4KBA.......... 5 GT61.......... 99
AC30 AUDIO INTERFACE...........79.50
PP40 PRINTER.................... 250
CT 1024 TERMINAL KIT............ 275
CTP.........15.50 CTS.........39.95
ALL SWTPC UNITS ARE KITS

SMOKE SIGNAL BROADCAST 16K RAM 595

LEAR SIEGLER ADM-3A W/ Cursor Control

- 12" CRT
- 24 LN X 80 CHAR
- RS232
- 20 mA LOOP

$875

SCAMP KIT................$99

FROM NATIONAL SEMICONDUCTOR
KEYBOARD KIT........... 95

★ BUILT UNITS ★

ICOM MICROFLOPPIES

PLUG COMPATIBLE FOR:
$100 BUS...FD2411..... **$1095**
SINGLE DRIVE FD2402......$ 649
ICOM FLOPPIES:
FF36-1 FRUGAL..........$1195
FF36-2 DUAL FRUGAL..... 1895
360-58 BLT;INTFC 8080.. 300
S171 POWER SUPPLY ... 250
FD360-2-5DUAL SYSTEM... 3000

KIM-1 6502...............$ 245

KIM-2 4K................. 179
KIM-3 8K................. 289
MANUALS PACKAGE.......... 15

TARBELL AUDIO CASSETTE KIT 120

INTERSIL INTERCEPT JR. $281

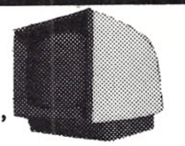
12K RAM................ 145
ROM/PROM BOARD.........74.65
YOU ADD MEMORY CHIPS
SERIAL I/O.............81.50
AUDIO VISUAL BOARD..... 125

SPECIAL DISCOUNTS! ON KITS & ASSEMBLED UNITS

SAVE UP TO 20% OFF KIT PRICE WHEN A PERIPHERAL IS PURCHASED AT THE SAME TIME. ($200 MAXIMUM DISCOUNT)

PERIPHERAL OVER $900 ➡ 20% OFF KIT
PERIPHERAL OVER $250 ➡ 10% OFF KIT
PERIPHERAL OVER $95 ➡ 5% OFF KIT

GREEN PHOSPHOR VIDEO MONITOR $150 +25 SHIPPING



STANDARD 1V P TO P COMPOSITE VIDEO INPUT, 16 MHz BAND WIDTH, RASTER SCAN 12x12x13" WITH POWER SUPPLY, VIDEO AMPLIFIER, DRIVING CIRCUITRY VENTILATION MUFFIN FANS, 7x9" HORIZONTAL VIEWING, CAPABLE OF 24 LINES x 80 CHAR., ANTIGLARE ¼" ETCHED GRADIENT DENSITY FACE PLATE, P39 GREEN PHOSPHOR, ON/OFF BRIGHTNESS CONTROLS, 115Vac, 60 W ... TRULY A COMMERCIAL UNIT BUILT TO WORK IN A DEMANDING ENVIRONMENT.

DATAPOINT 3300-200 $475 + $25 SHIPPING

THERMAL PRINTER

PARALLEL PRINTER WITH ADDITIONAL CIRCUIT BOARDS TO PROVIDE SERIAL INTERFACE, PRINTS UP TO 30 CPS, 100Vac PS USES WIDELY AVAILABLE NCR PAPER, 96 CH, ASCII, 80 COL, CRT COMPATIBLE 5x7 DOT MATRIX, SOLID STATE WITH LESS THAN 25 MOVING PARTS.

DATAPOINT CONSOLE PRINTER

BOTH UNIVAC AND SINGER BUILT THESE PRINTER MECHANISMS WHICH OPERATE AT 30 CPS FROM A ROTATING WHEEL. 65 CHAR. USES STANDARD FORM FEED PAPER. PINWHEEL IS INTERCHANGEABLE.

$395 +1285 lb. SHIPPING

UNIVAC 0769-06 PRINTER MECHANISM ONLY ... $195
INCLUDES MOTOR/PRINT WHEEL +SHIPPING 75lb.

KLEINSCHMIDT 311..... $250 + 75 lb SHIPPING

THIS 30 CH/SEC DRUM PRINTER SITS IN A SOUND-PROOF ENCLOSURE. 64 CHAR, PARALLEL INPUT, 80 CHAR/LN

TECHTRAN 4100........$595 + 35 lb SHIPPING

TAPE CASSETTE DRIVE. CAN RUN DIRECTLY FROM TERMINAL INDEPENDENT OF CPU. FULL EDIT CAPABILITY.

TO ORDER EQUIPMENT

1. ENCLOSE CHECK FOR FULL PRICE PLUS SHIPPING CHARGES (KITS - ADD $5 IF UNDER $100; $10 IF OVER) BANKAMERICARD & MASTERCHARGE ACCEPTED SEND CARD NUMBER, EXPIRATION DATE, INTERBANK #
2. CLEARLY IDENTIFY SHIPPING ADDRESS
3. DESCRIBE ITEM BY MODEL NUMBER

ALL MERCHANDISE WARRANTEED

SEND $1 FOR OUR CATALOG

INCLUDES DESCRIPTIONS OF OUR COMPLETE LINE OF KITS & UNITS, REVIEWS OF OVER 150 BOOKS, LIST OF NEW & SURPLUS PARTS AND ARTICLES - "ALL ABOUT PERSONAL COMPUTERS"

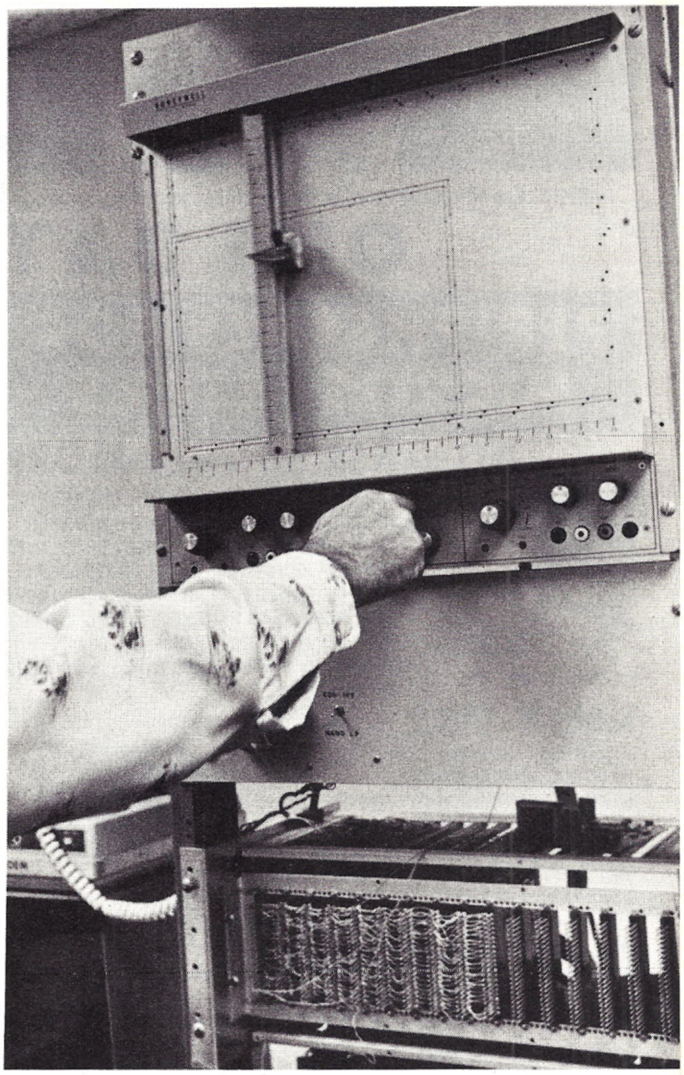
C27

Don's background: at Sperry Rand he'd been responsible for the interface between the IBM 1130 and an electro-optical laser scan. He'd built the interface between two computers. And he built his own computer at home for kicks. Don got the job.

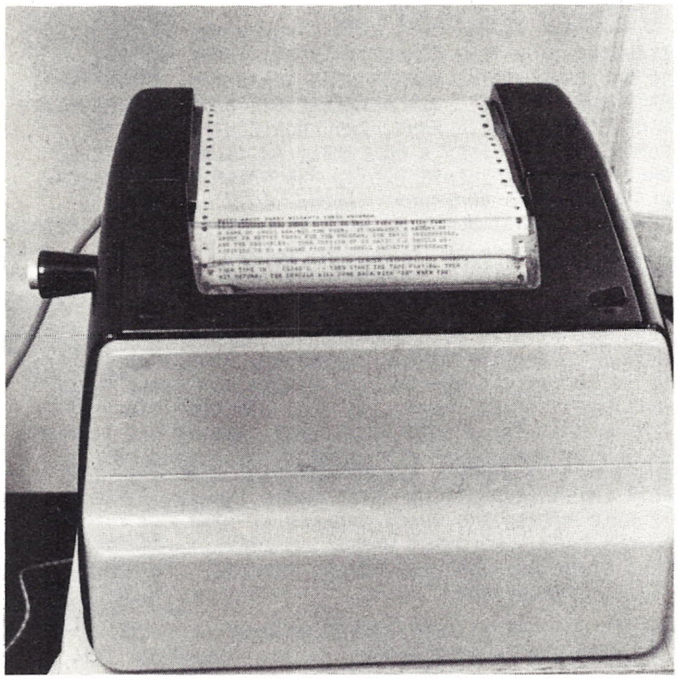
It was 1973, at Hughes, in Southern California. Don's new job was more computer-related than his former job had been. Though much of his work was classified, he could relate some work he'd done with a PDP-11, saying it was "a very nice machine with a very professional instruction set." As a matter of fact, Don claims to be something of a connoisseur of instruction sets, having written one for his machine. In order to do so, he explored all then-existing instruction sets in order to pull out those he preferred for his own machine.

His job involved a great deal of programming and hardware design. He preferred working on the PDP-11 over other computers being used at that time. It was used to control a Gerber plotter for which Don designed interface hardware and wrote software. Some of his other accomplishments include rewriting an editor, assembler, and operating system for the PDP-11. He professes to have had a lot of fun doing all of it.

When asked if he considered himself more of a hardware type or a software type, he replied, "I'm getting to the point now where it's sort of 50-50. My main concentration lately has been in software, but my education is in hardware. I feel I have a good mix of both." As a matter of fact, his background education and on-the-job experience are heavy in both, so his response seemed a bit modest.



Don demonstrates how the plotter works, either via BASIC or direct control. He'd initially intended its use for plotting diagrams but hasn't yet written the software. He does have software that draws letters graphically and controls their size.



Printer was once a 32 TTY donated by a friend. Don converted it to a 33 and now uses it to type operator manual pages for his cassette interface.

Cassette Interface Evolution

The chain of events which led Don to develop, then market his cassette interface are many and roundabout. One of the influencing factors was frustration, as related in this tale of woe: "I picked up an 8-track cartridge player from Lockheed and mounted it in a panel, put some lights and switches on it and built an interface – while back in Huntsville. It worked for quite awhile . . . then started flaking out. It used two tracks – one for clock and one for data. If the heads got sloppy, it wouldn't clock right. Once I lost a whole editor and had to key the whole thing in again through the front panel switches."

Frustrated, he switched to hard disk. Finally, he designed a cassette interface for the same computer. His decision to use cassette tape for storage was a matter of economics. It was the least expensive route, and at Sperry Rand he'd discovered that a cheap recorder would work fine. The discovery was made when he was keeping track of tape recorders for use in space flights. They used the phase encoding system, packing about 1,000 bits per inch. Since the recorder was to be used in Skylab, it had to withstand vibrations, and Don's job was to test them. They had highly sophisticated equipment to check random patterns, check them back for errors, and so forth. A co-worker suggested they try the test on a Sony tape recorder. They did, and Don says, "We couldn't believe it worked!" The recorders they were buying for the test cost several thousand dollars and they were astonished that the little Sony would do the same thing. Since he then had his

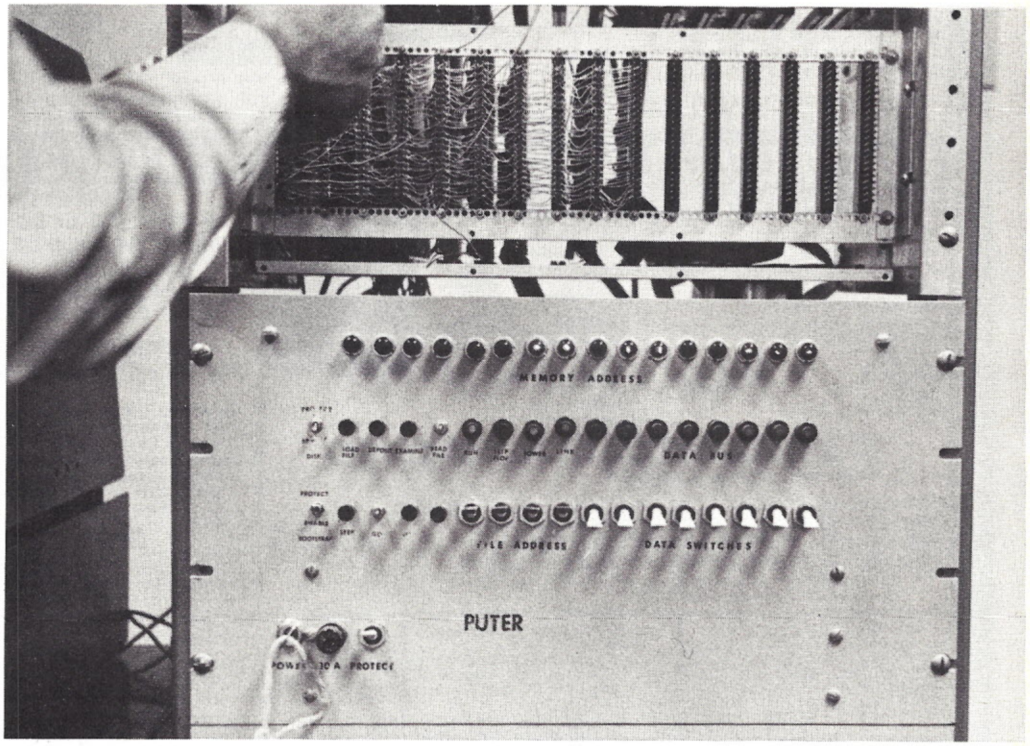
computer working at home, Don designed an interface to do the phase encoding and self clocking. It was successful, and he's still using it.

Beginning of SCCS

Using a list of people Don had contacted through the Amateur Computer Society, he kept in touch with those who were close to his home in Southern California. It was through one of those friends that he eventually met Judge Pearce Young. Pearce suggested that a meeting be called of all the people on the list to exchange information and views on computers. Together they discussed the possibility of a club with others to see if there were enough interest and batted the idea around for a couple of months. Then the Mitsmobile (Altair) came to Van Nuys, California and the microcomputer revolution was out in the open. More names and addresses were gathered and an announcement was sent out to about one hundred people for a meeting to be held at Don's apartment building. Two hundred people showed up. Two or three brought Altairs and one person demonstrated a PDP-8. So, on June 15, 1975, the first meeting was held for what became The Southern California Computer Society.

Tarbell's Home Brew System

The electronics of his computer consists almost entirely of TTL circuits, with a few RTL and DTL circuits "thrown in." Interfaces were built and installed to handle the CRT, a modem, the cassette recorder, disk controller, TTY, and the plotter. The modem, though not now in use, enables Don's computer to talk to other computers. He once had it set up to call his computer from work so he wouldn't have to stop working on it. All of this was completed and working before Don arrived in Southern California in 1973. Since then he's worked on the disk system and software.



Front panel switches came in handy when Don once lost an entire editor using a faulty 8-track cartridge player. He keyed the entire program in again, then switched to cassette. Above the switch panel is the interface card rack to handle all I/O devices. This computer knows who it is . . . when SPL is running ASK and we ask the computer WHAT IS YOUR NAME, it of course responds, PUTER.

The software he's written includes the bootstrap, disk monitor, editor and conversational assembler that works with the CRT and the hard disk. He's also written his own version of BASIC and describes it as being very much like Tiny BASIC but not as nice as Altair BASIC. It does have features however that others do not. For one, it will perform infinite precision calculations. For instance, a 1,000 digit number may be divided by a 500 digit number in about four hours. The printed result may be limited by using the LIMIT command:

LIMIT TO n

It will also handle long variable names and can use symbolic labels rather than be restricted by line numbers. These two features tend to make programming and reading programs much easier. For use with limited memory, a CHAIN command automatically brings in subsequent portions of data from disk. The disk monitor incorporates extremely easy to use commands. For instance, when in the bootstrap, "C" calls up the cassette file or "D" brings in files from disk. The bootstrap, incidentally, is in core memory. The disk operating system gives complete sentences describing file names, lest we forget why we designated a specific file or what it was. If an error is typed, it is described in English. It will edit, save files on disk, or print on hard copy. It permits moving from one program to another without having to go in and out of the monitor. It controls the hard copy I/O so that the printer may be stopped at any point in motion. The editor will do the usual edit functions found in similar editors and some not found in most. For instance, it can locate all occurrences of a character or word in a program.

Another of Don's feats is a sentence processing language he calls SPL. It works very much like BASIC except it does a string, rather than a numerical operation. A sentence typed in normal English is scanned using a phrase-matching process and comes back with English sentence responses.

Using a program called ASK, we interrogated the computer. Here's the result of some brief questioning:

Q: How am I today?
A: Today is a holiday

Q: What is tomorrow?
A: What does it matter

Q: When will I be rich and famous?
A: When is the end of time

You gotta watch it with some of these computers. ■

DIP PLUGS AND COVERS

Use for mounting diodes, resistors, jumpers, etc.
Gold plated parts for long wear.

	PLUGS 1-24	25-49	50-99	COVERS 1-24	25-99
8 pin	.46	.39	.32	.10	.09
14 pin	.48	.40	.34	.10	.09
16 pin	.56	.47	.38	.10	.09
24 pin	.79	.72	.65	.15	.13
40 pin	$1.23	$1.08	.93	.25	.22

DIP PLUG INTERCONNECTS

Ideal for use from board to board, remote switches, in test equipment, lamp panels, etc. Made from gold plated connectors, color coded ribbon cable, molded plugs.
Very flexible and durable.
To order, make up part number from characteristic and find price in chart. For instance: S-14P-18 is single ended 14 pin interconnect 18 inches in length. Price is $1.72. D-24P-06 is double ended 24 pin, 6 inches in length. Price is $4.55. Quantity pricing is available.

PRICE CHARTS

No Of Pins	6"	12"	18"	24"	36"	48"
SINGLE END						
14P	1.51	1.62	1.72	1.83	2.05	2.26
16P	1.64	1.76	1.87	1.99	2.21	2.44
24P	2.49	2.69	2.88	3.08	3.48	3.87
DOUBLE END						
14P	2.76	2.87	2.97	3.08	3.30	3.51
16P	3.01	3.13	3.24	3.36	3.58	3.81
24P	4.55	4.75	4.94	5.14	5.54	5.93

PRESTRIPPED WIRE WRAP WIRE

Highest quality 30 ga. Kynar insulated silver plated wire for wrapping. Stripped 1" on both ends. Indicated lengths are lengths of insulated portion. Packed 100 per sturdy plastic vial or 1000 per poly bag. Compare our prices!!!. Available in Black, Red, Yellow and Green. State color desired.

Length	Price per tube of 100	Price per bag of 1000
1"	$1.48 (WW30VC-1)	$11.84 (#WW30BK-1)
2"	$1.60 (WW30VC-2)	$12.80 (#WW30BK-2)
4"	$1.85 (WW30VC-4)	$14.80 (#WW30BK-4)
6"	$2.20 (WW30VC-6)	$17.60 (#WW30BK-6)

Dealer Inquiry Invited

ROLLS OF WIRE SAME AS ABOVE (30 ga. KYNAR)
100ft Roll in poly bag...$3.45 10 Rolls, mixed....$24.00

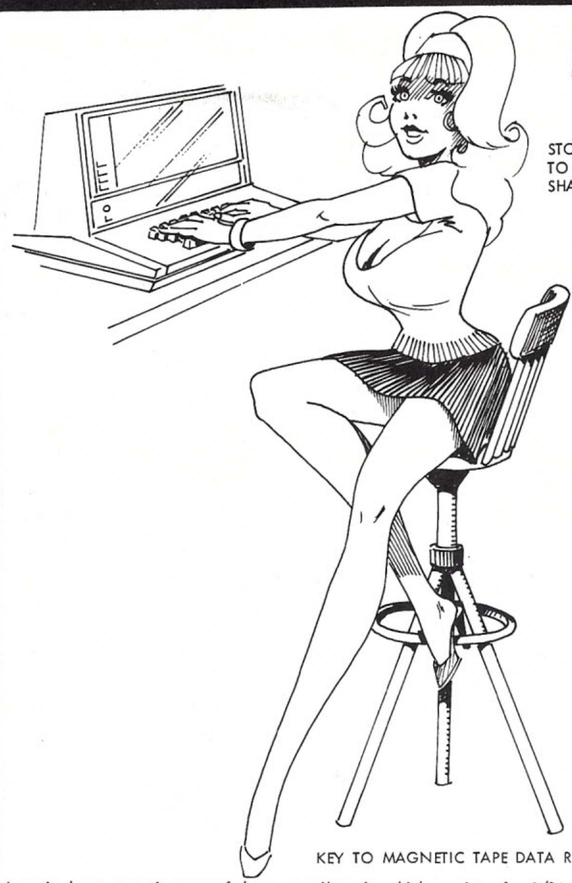
WIRE WRAP I.C. SOCKETS, 3 LEVEL GOLD
14 pin 38¢ ea, 10 for $3.60 16 pin 42¢ ea, 10 for $3.90

HIGH CURRENT STUD DIODES

D2131 200V, 25A.................................. $ .85
D2135 400V, 25A.................................. $1.00
D2138 600V, 25A.................................. $1.55
3289 200V, 100A.................................. $5.85

Contract Cancellation Specials.....All Full Spec, NEW!!

SN7490.......39¢	2N3734.....38¢	2N3773 $1.75
LM320H-12...$1.29	2N5861.....75¢	2N4401..5/$1
1N5231A.... 25¢	1N3572.....90¢	2N4403..5/$1
1N5226B..... 35¢	G.E.D45C5 44¢	2N1557..$1.39
741(mini-dip..3/$1	50V, 3Amp Epoxy Bridge........ 79¢	



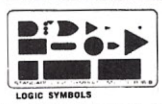
AMP L'ANNY
Says

STORE A WORLD OF DATA ON THE KEY TO TAPE MACHINE FROM TRI-TEK AND SHARE A BYTE WITH A FRIEND!!!!!!!!!

KEY TO MAGNETIC TAPE DATA RECORDERS

Anny is shown operating one of these versatile units which consists of a 1/2" magnetic tape recorder sitting on top of keyboard/controller/display module. These units were made by PERTEC, one of the most respected names in data recording and were used to replace punched card input. They are of late design and recent manufacture. From the operators chair, it is just like a key-punch. Instead of data going into cards in Hollerith, it goes onto Mag tape in EBCDIC. They may be used for that purpose or the tape drive can be separated, control and data lines brought out for use directly on your system. Has internal memory/buffer for 80 or 200 character storage. Display panel indicates character, character number, record number. Read back circuits allows search on record key, editing, duplicating...! These are not obsolete!!! A giant factory closing brings you these at about 5¢ on the dollar. All units are complete and in good condition. They have not been functionally tested but have been inspected for damage. All are sold on as is basis. There are national service shops for these units and parts are available. Hundreds of these units are being used right now in business and industry.
Prices are as is (complete and inspected), F.O.B. our warehouse in Rochester, N.Y. and shipped by truck, freight collect only. Unit weight is approximately 70 lbs. Comes in 7 or 9 track versions. Model 4311 has remote data communication channel. All units, less tape.

KT-4301-7......... 7 track data recorder.......$249.00
KT-4311-7 .... 7 track with remote $299.00
KT-4301-9......... 9 track data recorder....... $329.00
Operators and maintenance manuals(sold separately only)$20



PROFFESSIONAL LOGIC SYMBOL TEMPLATE
MIL-STD 806-B (Half Size)................$3.25

Fast Signal Diode
115V VR 100mA If. Reverse recovery time is less than 20 nS at 100mA forward! 6 pf cap. Same size as 1N914, 1N4148. D600................20/$1

TO-5 Heat Sink. THERMALLOY 2211B 2-piece black anodized for maximum heat dissipation. HS2211B. 5/$1

6.8V, 50Watt Zener. Made by Motorola in TO-3 case. Gold plated. $1.00

NEW BOOK FROM NATIONAL
MOS LSI. Giant data book filled with spec and apps on large scale MOS circuits from National Semiconductor Corporation. Price includes shipping in U.S. only...$4.25 Outside U.S., add postage for 2 lbs.

MCM6571 Full ASCII Character Generator....... $9.95

DATA BOOKS BY NATIONAL SEMICONDUCTOR

DIGITAL. Covers TTL, DTL, Tri-State, etc.......$3.95
LINEAR. Covers amplifiers, pre-amps, op-amps... $4.25
LINEAR APPLICATIONS VOLUME I. Dozens of application notes and technical briefs covering the use of op-amps, regulators, phase locked loops and audio amps.................................... $3.25
LINEAR APPLICATIONS VOLUME II. Takes up where Volume I left you--All the latest linear devices. Along with Vol I you have a great source of application data on the most widely used devices as well as new types just appearing.....................$3.25
CMOS Gates, Flip Flops, registers, etc........$3.00
VOLTAGE REGULATORS. A must for anyone making a power supply. Complete theory including transformers, filters, heat sinks, regulators etc........... $3.00
MEMORY. Info on MOS and Bipolar memories, RAMS ROMS, PROMS and decoder/encoders........... $3.95
INTERFACE. Covers peripheral drivers, level translators, line driver/receivers, memory and clock drivers, sense amps, display driver and opto-couplers..... $3.95
SPECIAL FUNCTIONS DATA BOOK. Contains detailed information for specifying and applying special amplifiers, buffers, clock drivers, analog switches and D/A-A/D converter products.................. $3.25
AUDIO HANDBOOK Contains detailed discussions, including complete design particulars, covering many areas of audio with real world design examples... $3.25

TRI-TEK, INC.
6522 NORTH 43RD AVENUE,
GLENDALE, ARIZONA 85301
PHONE 602 - 931-6949

We pay surface shipping on all orders over $10 US, $15 foreign in US funds. Please add extra for first class or air mail. Excess will be refunded. Orders under $10, add $1 handling. Please add 50¢ insurance. Master charge and Bank America cards welcome, ($20 minimum). Telephone orders may be placed 10AM to 5:30PM daily, Mon thru Fri. Call 602-931-4528. Check reader service card or send stamp for our latest flyers packed with new and surplus electronic components.

RONDURE COMPANY

NEW ADDRESS
2522 BUTLER
DALLAS, TEXAS 75235
Phone: (214) 630-4621

the computer room

Where We Ship from Inventory the Same Day Your Order Arrives*

A SELECTRIC TERMINAL COMPLETE WITH RS-232/C INTERFACE AND CERTIFIED FOR MAINTENANCE BY A NATIONAL SERVICE COMPANY.

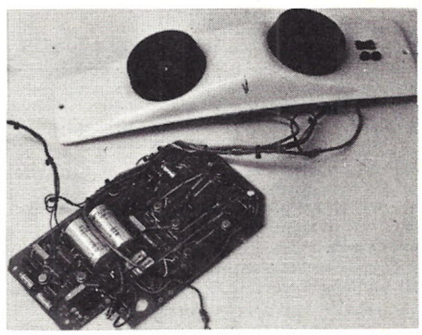
$895.00

*Maintenance limited to cities in which service now offered. Shipped the same day as certified check or money order arrives. When regular checks accompany order, equipment is shipped when regular check clears.

Specifications

- ☐ Size: 21" wide x 21" deep x 8" high.
- ☐ Power Input: 115 Volt, 60 Hz.
- ☐ Mounting: Tabletop.
- ☐ Interface: RS232C. (BCD code)
- ☐ Weight: 54 lbs.
- • Documentation $20 with unit.
- • ASCII conversion available.
- • Shipping – collect

CANNON 25 PIN RS232 PLUGS $1.50 each Min. order 10
MODEL 33 KSR TELETYPES w/coupler $595.00
MODEL 38 ASR ... NEW $1095.00
12" MOTOROLA MONITOR (20 MHz Bandwidth, 800 lines) – NEW $160.00
LINE PRINTERS – USED – 150 1pm $895
LINE PRINTERS – USED – 480 1pm $995.00

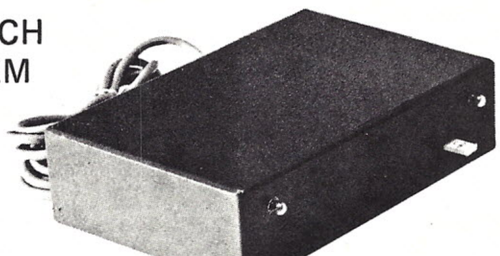


SMALL ITEMS

TERMS: Check or Money Order. Add $2.00 shipping and handling. All others shipping packaging and shipping collect.

CARTERFONE MODEL 318

ASYNCH MODEM



- HARD WIRE
- TTY OR RS-232B INTERFACE
- ORIGINATE ONLY
- UP TO 300 BPS

USED – UNTESTED .. $25.00
USED – TESTED $80.00
We ship prints with these.

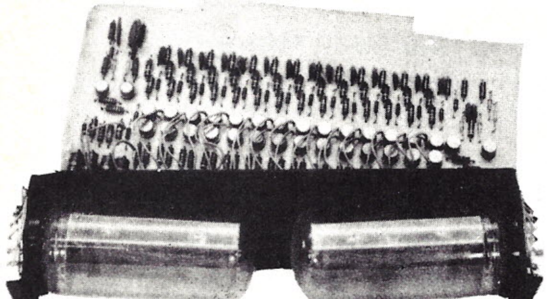
ACCOUSTICAL MODEMS – ORIGINATE ONLY
USED – UNTESTED
Physically fit into Model 33 Teletype. Manufactured by Paragon partial documentation 2 for $25

ACCOUSTICAL MODEMS – ORIGINATE ONLY USED – UNTESTED
$20.00 ea. 2 for $35

NO EQUIPMENT INCLUDES PRINTS OR DOCUMENTATION (unless stated).
NO CONNECTING CORDS OR CONNECTORS.

RONDURE COMPANY 2522 BUTLER
DALLAS, TEXAS 75235
PHONE: (214) 630-4621

R-7



Giant Alpha—Numeric NIXIE Tubes

One of the ultimate forms of computer output. These Burroughs NIXIE tubes have characters 2½" high, easily readable from long distances. The character is a 15 segment Alpha-Numeric type. The tubes (no. B7971) are mounted in 2 sockets on a PC board, which contains 33 driver transistors. We supply data on the tubes. Ideal for large clocks, store displays, sports scoreboards, or any other alpha-numeric display. Shipping weight 2 lbs.

STOCK NO. B6001 set of 2 tubes, sockets, PC board $6.95, 2 sets/12.00

GENERAL PURPOSE LOCK

Keep your valuable equipment locked up. This general purpose lock is very easy to use; mounts in ¾" dia hole, up to ½" thick. The "hook" catches on any ¼" bolt. 180 throw. Complete with 2 keys. Extra keys .25 each.

STOCK NO. B5384 $1.50/set, 4/5.00

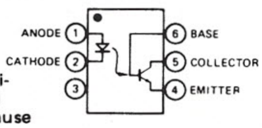
Arrow-Hart

LOCKING SWITCH SPST

A SPST lock switch made by Arrow—Hart. Useful as a main power switch, write protect switch, burglar alarm switch, etc. Mounts in ½" hole. With key, mounting nut, and attractive beveled bezel.

STOCK NO. B5294 $1.50 each, 4/5.00

MOTOROLA OPTICAL COUPLER

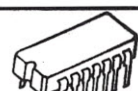


Optical coupler in a 6 pin mini-DIP. GaAs diode source, NPN phototransistor receiver. Because of their almost infinite isolation between input & output, optical couplers make ideal buffers. They can be used to control AC with TTL (through SCRs or triacs), switch high voltage with TTL, as line receivers, as interface circuits between logic families, etc.

STOCK NO. B4524 $1.00 each, 6/5.00

GRIGSBY BARTON

DRY REED RELAY



Type G821 A SPST-14 pin, 24vdc 2150 ohms. Sheilded diode non-electrostatic sheild. They will fit standard 14-pin receptacles, they are designed for direct printed circuit board mounting where desirable. Internal clamp in circuit Bd mounting where it is desirable. Internal clamp diodes for protection of drivers by suppressing back EMF There is data available.

STOCK NO.. 1003 B $1.75 2/ 3.00

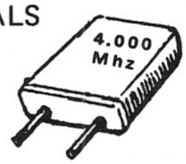
P.C. ROTARY SWITCHES

Half inch switches designed with only two moving parts. Mount directly on Printed Circuit Boards or panels. Positive detent, 36° indexing stops. 250 mA at 28vdc. 4,000 switching positions per position-contact resistance 0.05-ohm max. Operation temperature -65° C to - 150° C.

87-22-25 Knob Adjust 0.555 dia x 0.280 high
2 pole - 5 position.
STOCK NO. 1004 B $2.00 4/ 7.00

87-12-10 Screw driver Adjust 0.500 dia x 0.190 high
Single pole 10 position
STOCK NO. 1005 B $1.75 2/$3.00

CLOCK and TIME BASE CRYSTALS

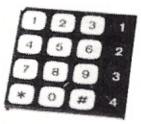


STOCK NO.K1000 4.000 Mhz Crystal HC/18 holder, pin leads

STOCK NO.K1008 5.000 Mhz Crystal HC/18 holder, wire leads.

Either crystal $3.95 2/7.00

16 KEY TOUCH TONE PAD

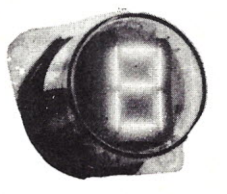


This 16 terminal TOUCH TONE pad was made for RAYTHEON. The keys are all SPST switches, brought out to 2 sets of terminals of 8 an; 9 pins, with .1 spacing so that they fir MOLEX IC terminals. The pad is 3" square. Individually packed with pin connection date. Ideal for touch tone uses. STOCK NO.K5488 $6.95 ea. 2/12.00

GOLD PLATED MOLEX TERMINALS

Gold plated MOLEX terminal strips in sections of 8 terminals. Any number of terminals may be obtained by cutting or adding sections. STOCK NO. M7490 12 8 terminal pieces 1.49, 48/5.00, 96/8.00

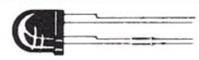
RCA End View NUMITRONS



The RCA end view NUMITRON is a 7 segment incandescent readout tube, with a character height of 5/8". It requires a 9 pin Novar socket, which we furnish with each tube. The tube operates at a nominal voltage of 4.5v, and draws 24 ma. per segment. It can be filtered to any color. With data.

STOCK NO. B5207 with sockets $2.75 ea, 4/10.00, 8/18.00

OPCOA RED LEDs



OPCOA no. OSL-3-30 red LED. All are new & tested good. Use in all types of displays. .20" dia., can be panel or PC board mounted.

STOCK NO. B4536 .25 each, 5/$1.00

DIODE-LITE™ LED LOGIC/STATE FAULT INDICATORS

DIALIGHT 555-2003 GaAs LED indicator, red diffused light. Built in series resistor to make it directly usable with TTL; draws 6 ma. @ 5 volts. .10" wide x .24" deep x .25" high. Vert, PC mount.

STOCK NO. B4537 .50 each, 5/$2.00

100 WATT ISOLATION TRANSFORMER

100 watts, continuous operation. 115 volts to 115 volts.
2 7/8"x3 3/8"x3½" 7 Lbs.
STOCK No.K9971 $5.95 ea. 2/11.00

HIGH CURRENT TRANSFORMER

This transformer will deliver 1 of the following : 15 V.@ 15 A., 12 V. @ 15 A., 9 V. @ 10 A, 6 V. @ 8 A.or 3 V @ 6 A. 7 lbs.
STOCK NO.K6481 $8.95 ea. 2/16.00

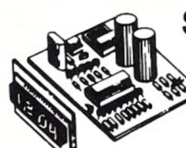
Send for our latest free catalog. We welcome Master Charge & BankAmericard orders. Minimum credit card order $15.00. Please include ALL numbers for proper processing. Include sufficient postage. Excess is refunded. Minimum order $5.00

DELTA ELECTRONICS CO

POST OFFICE BOX 2, AMESBURY, MASS. 01913 Phone (617) 388-4705 D13

S.D. SALES CO. P.O. BOX 28810 – Ⓚ DALLAS, TEXAS 75228

JUMBO LED CAR CLOCK



$16.95
KIT

Alarm Option – $1.50
AC XFMR – $1.50

THE HOTTEST SELLING KIT WE EVER PRODUCED!

You requested it! Our first D.C. operated clock kit. Professionally engineered from scratch. Not a makeshift kluge as sold by others.

Features:

- A. **Bowmar Jumbo –.5 inch LED array.**
- B. **MOSTEK – 50250 – Super Clock Chip.**
- C. **On board precision crystal time base.**
- D. **12 or 24 Hr. Real Time Format.**
- E. **Perfect for cars, boats, vans, etc.**
- F. **P.C. Board and all parts (less case) included.**

50,000 SATISFIED CLOCK KIT CUSTOMERS CANNOT BE WRONG!

THIS MONTH'S SPECIALS

AMD – 8080A	$14.95
Z-80 CPU	49.95
82S129 1K PROM	2.50

1702A 2K EPROM

We tell it like it is. We could have said these were factory new, but here is the straight scoop. We bought a load of new computer gear that contained a quantity of 1702 A's in sockets. We carefully removed the parts, verified their quality, and are offering them on one heck of a deal. First come, first served. Satisfaction guaranteed! U.V. Eraseable. **$6.95 ea. 4/$25**

UP YOUR COMPUTER!
21L02-1 1K LOW POWER 500 NS STATIC RAM

Time is of the essence! And so is power. Not only are our RAM's faster than a speeding bullet but they are now very low power. We are pleased to offer prime new 21L02–1 low power and super fast RAM's. Allows you to STRETCH your power supply farther and at the same time keep the wait light off. **8 for $12.95**

60 HZ CRYSTAL TIME BASE
S.D. SALES EXCLUSIVE!

$5.95 ea. **2/$10.00**

KIT FEATURES:

- A. 60HZ output with accuracy comparable to a digital watch.
- B. Directly interfaces with all MOS clock chips.
- C. Super low power consumption (1.5 MA typ.)
- D. Uses latest MOS 17 stage divider IC.
- E. Eliminates forever the problem of AC line glitches.
- F. Perfect for cars, boats, campers, or even for portable clocks at ham field days.
- G. Small size; can be used in existing enclosures.

Kit includes Crystal, Driver IC, PC board, plus all necessary parts and specs. At last count – over 20,000 sold!

S.D. SALES EXCLUSIVE
$12.95 MOS 6 DIGIT UP–DOWN COUNTER $12.95

40 PIN DIP. Everything you ever wanted in a counter chip. Features: Direct LED segment drive, single power supply (12 VDC TYPE.), six decades up/down, pre-loadable counter, separate pre-loadable compare register with compare output, BCD and seven segment outputs, internal scan oscillator, CMOS compatible, leading zero blanking. 1MHZ. count input frequency. Very limited quantity! WITH DATA SHEET

7400–19c	7411–29c	7451–19c	7490–65c	74153–75c
74LS00–49c	7413–50c	7453–19c	74LS90–95c	74154-1.00
7402–19c	7416–69c	7473–39c	7492–75c	74157–75c
74LS02–49c	7420–19c	7474–35c	7493–69c	74161–95c
7404–19c	7430–19c	74LS74-59c	7495–75c	74164-1.10
74L04–29c	7432–34c	7475–69c	7496–89c	74165-1.10
74S04–44c	7437–39c	7476–35c	74121-38c	74174–95c
74LS04-49c	7438–39c	7480–49c	74123-65c	74181-2.50
7406–29c	7440–19c	7483–95c	74132-1.70	74191-1.25
7408–19c	7447–85c	7485–95c	74S138-1.95	74192-1.25
7410–19c	7448–85c	7486–45c	74141-75c	74193-1.00
				74195–69c

TTL INTEGRATED CIRCUITS

1000 MFD Filter Caps

Rated 35 WVDC Upright style with PC leads. Most popular value for hobbyists. Compare at up to $1.19 ea. from franchise type electronic parts stores. S.D. Special 4/$1.

Slide Switch Assortment

Our best seller. Includes miniature and standard sizes; single and multi-position units. All new, first name brand. Try one package and you'll reorder more! Special **12/$1.00**

RESISTOR ASSORTMENT

¼W 5% & 10% PC leads. A good mix of values. **200/$2.**

P.C. LEAD DIODES

1N4148/1N914 100/$2.00
1N4002-1A. 100 PIV 40/$1.

HEAVY DUTY
Full Wave Bridge
25 AMP 50 PIV
$1.25

Disc Cap Assortment

PC leads. At least 10 different values. Includes .001, .01, .05, plus other standard values. 60/$1.00

$9.95 KIT

P.C. Board – 3.00
AC XFMR – 1.50

Do not confuse with Non-Alarm kits sold by our competition! Eliminate the hassle – avoid the 5314!

SIX DIGIT ALARM CLOCK KIT

We made a fantastic kit even better. Redesigned to take advantage of the latest advances in I.C. clock technology. Features: Litronix Dual ½" displays, Mostek 50250 super clock chip, single I.C. segment driver, SCR digit drivers. Greatly simplified construction. More reliable and easier to build. Kit includes all necessary parts (except case). P.C.B. or XFMR optional. *NEW! WITH JUMBO LED READOUTS!*

Motorola SCR

2N4443. 8 AMP 400 PIV. P.C. Leads **3/$1.**

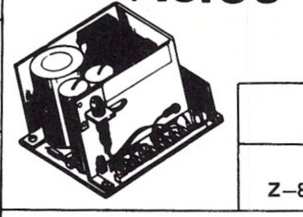
FAIRCHILD - TBA 641

4W. Audio power Amp. Just out! In special heat sink DIP. One super audio IC. **$1.50 with data**

FND-359 -Led Readout

.4 IN. Common Cathode. High effeciency. Has FND-70 PIN OUT. **59c**

$15.95



COMPUTER POWER SUPPLY

A very fortunate purchase. One of the best industrial quality REGULATED supplies we have seen. High performance, small size. Input is 120 VAC 60 HZ. Has the following regulated outputs: **–5VDC@800MA; –15VDC @ 1.25 AMP; –25VDC @ 180 MA.** Sold at a fraction of original cost. Do yourself a favor and order NOW. We expect a quick sellout.

NEXT MONTH:

S.D. will have music for your ears. Watch our ads.

For your Imsai or Altair 8080 Computer:
Z–80 CPU Kit – $149. 4K Low Power Ram Kit – $89.95

OUR CATALOG

is chocked full of rare parts bargains, deals, RAM or CPU kits, plus much more. Yours FREE!

PRICES SHOWN SUBJECT TO CHANGE WITHOUT NOTICE.

Terms: Money back guarantee. No COD. Texas residents add 5% sales tax. Add 5% of order for postage & handling. Orders under $10. add 75c. Foreign orders: US funds only!

Call your Bankamericard or Master Charge order in on our continental United States toll free Watts:

1–800–527-3460

Texas Residents Call Collect:

214/271-0022

Special Thanks to:
Dennis, Fred, Abe, Bill, Sam, Hal, Tom, Alex, John, Ely, and Larry

S.D. SALES CO.
P. O. BOX 28810 Ⓚ
Dallas, Texas 75228

ORDERS OVER $15.00 – CHOOSE $1.00 FREE MERCHANDISE

computer display terminal

This display terminal has an integral controller B/W cathode ray rube and keyboard. The system has a serial I/O interface for communication and I/O interface for a printer.

External logic & power pack not shown.

DISPLAY (P/N 4802-1095-501) FEATURES:
- 17" B/W CRT
- 41 lines of data
- 52 characters per line
- Characters are generated by a diode matrix "graphic" technique
- 21 special push-buttons wired for a program call up
- Brightness Control
- Self-contained power supply

KEYBOARD (P/N 4802-1115-501) FEATURES:
- Reed switch technology
- 54 data keys
- 28 special keys detachable with cable

LOGIC UNIT (P/N 4802-1157-502) FEATURES:
- 1024 by 6 bit core memory
- Printer I/O interface
- Communication I/O interface

POWER: 115V, 50/60 Hz, 500 Watts

WEIGHT: 210 lbs. (including logic unit, keyboard, display and cables.)

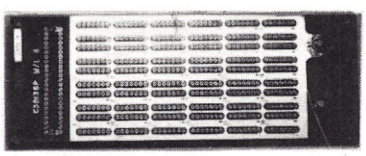
$ 180.00

4 way cursor control, graphics display.

The story: These are unused terminals made for airport ticketing & seat assignment. After several years of storage they require tinkering to make operable. We have some hints printed such as cleaning PC fingers. One of our customers has this tied into his KIM-1, another has his running with his IMSAI. We have data on this. Should be useable on most common computers. A hell of a deal and all for a paltry $180.00. Don't be left out as many were on our past VIATRON deal. Sold "as is" all sales final.

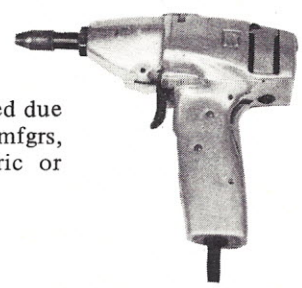
FOB LYNN MASS (you pay shipping)
Check with order please.

WITH COMPLETE DOCUMENTATION



Here is a real deal in a PC module 6x5 sockets (30). List price over $50 each, most by AUGAT some pre-wired. New, unused boxed, 14 or 16 pins 5x6 sockets. $15.00 each or 2 for $25, state your choice 14 or 16 pin.

WIRE WRAP GUNS



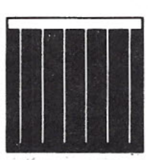
Used wire wrap guns, released due to factory closure. Various mfgrs, some Ingersol Rand. electric or air.
No collets. State choice.
Cost over $100.00 each.

Our price only $15.00 each.

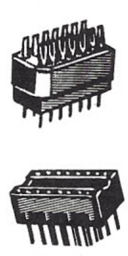
SOLAR CELLS

Designed for the space program, these are the highly efficient silicon high output cells. Used for powering equipment, charging batteries. Made by Ion Physics Corp. Each with spec sheet.
Size .394 x .788" 65 mA, .43 V
$1.25 12/$12.00
Size .788 x .788" 125 mA, .43 V
$1.60 12/$15.00

shown actual size



SILICON SOLAR CELL



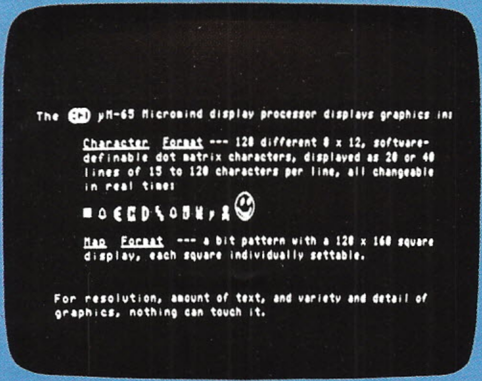
IC Sockets, while they last . . .

8 Pin	10/$1.00
14 Pin	10/$1.25
16 Pin	10/$1.50
18 Pin	10/$1.75
14 Pin IC connector	10/$1.25

Meshna

Please add shipping cost on above. Minimum order $10
FREE CATALOG SP-9 NOW READY
P.O. Box 62K, E. Lynn, Massachusetts 01904

M-2



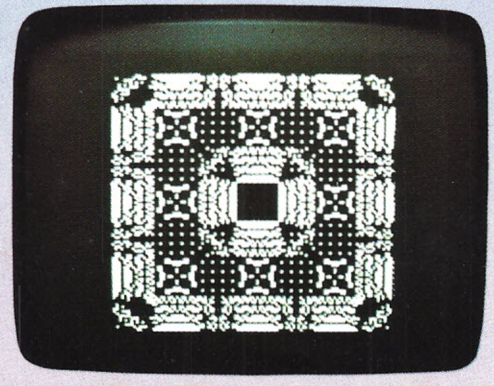
Key Into Maxi-Power @ Micro-Price

Micromind is an incredibly flexible, complete and expandable, hardware/software, general purpose computer system. You won't outgrow it.

Hardware includes an 80 key, software-definable keyboard, I/O interface board, 6500A-series microprocessor (powerful enough for advanced computing), a high-detail graphics and character display processor, power supply, rf modulator, and connections for up to 4 tape recorders plus TV or monitor. An interconnect bus permits 15 additional microprocessors, parallel processing and vastly increased computing power.

System software—including ECD's own notsoBASIC high level language, on advanced error-correcting tape cassettes—provides a word processing editor, a powerful assembler, a debugger, a file system, graphic routines, and peripheral handlers. We also include dynamic graphic games: Animated Spacewar and Life.

ECD's standard Micromind μM-65 supplies 8K bytes of memory. Additional 32K byte expansion boards and a mapping option give Micromind expandable access to 64 Megabytes. Utilizing software-controlled I/O channels, Micromind's advanced encoding techniques load data from ordinary tape recorders at 3200 bits per second.



Micromind comes to you ready-to-use, factory assembled and fully tested. Among microcomputers, it has the largest memory capacity and the fastest storage. You're looking at the work of the finest display processor on the market. You won't find a microcomputer with a more powerful CPU. You won't find a computer with a more flexible keyboard. You won't find anything to touch it at **$987.54.**

So, quit the kluge scene and key into Micromind. You'll be a main frame performer, with all the comforts of home. We're not fooling...this is the cat's μ!

ECD CORP.
196 Broadway, Cambridge, Mass. 02139
(617) 661-4400

ECD

Name ______________________

Address ______________________

City/State ______________ Zip ________

☐ Fantastic! Check enclosed: **$987.54.** Shipping paid by ECD

☐ BankAmericard ☐ Master Charge Mass. Resident add 5% Sales Tax

______________ Expiration Date ________

Signature ______________________

☐ Send me your brochure.

7

Actual unretouched photographs.

/ability.

It Comes Naturally With The Altair™ 8800b

The Altair 8800b from MITS: the second generation design of the microcomputer that started it all. The mainframe that has the abilities everyone is demanding from microcomputers today:

Expand-ability:

The Altair 8800b power supply and one-piece, 18-slot motherboard allow efficient and easy expandability for memory and I/O options. All Altair PC boards are designed to give you maximum capability/lowest power usage possible per board. This means that for each slot used you get more features and require less power, than with any of the "off-brand" Altair-bus-compatible boards.

Whether you buy an entire system up front or choose to expand gradually, it's easy to get the configuration you need with the complete family of Altair peripheral equipment, including floppy disk, line printer, audio cassette record interface, A/D converter, PROM programmer, serial and parallel I/O boards, choice of four different memory boards and many others.

Reli-ability:

The unique design features of the Altair 8800b, which have set the standard for the microcomputer industry, make it the most reliable unit of its kind. The Altair 100-pin bus, the now-standard design used by many imitators, has been "standard" all along at MITS. The unique Front Panel Interface Board on the Altair 8800b isolates and filters front panel noise before it can be transmitted to the bus. The all-new CPU board utilizes the 8080A microprocessor, Intel 8224 clock generator and 8216 bus drivers.

Flex-ability:

Meeting the diversified demands of an ever-increasing microprocessor market requires flexibility: not just hardware flexibility but software flexibility as well. MITS software, including the innovative Altair BASIC language, allows the full potential of the Altair 8800b computer to be realized.

8K ALTAIR BASIC has facilities for variable length strings with LEFT$, RIGHT$, and MID$ functions, a concatenation operator, and VAL AND STR$ functions to convert between strings and numbers.

Extended ALTAIR BASIC allows integer, single and double precision variables, automatic line numbering and renumbering, user-defined string functions, PRINT USING for formatted output and a powerful EDIT command for editing program files during or after entry. Extended statements and commands include IF . . . THEN . . . ELSE, LIST and DELETE program lines, SWAP variables and Trace On and Off for debugging.

Disk ALTAIR BASIC has all the features of Extended BASIC with the additional capability to maintain sequential and random access disk files. Utilities are provided for formatting disks and printing directories.

In all versions of ALTAIR BASIC you get the ease and efficiency of BASIC for the solution of real world problems.

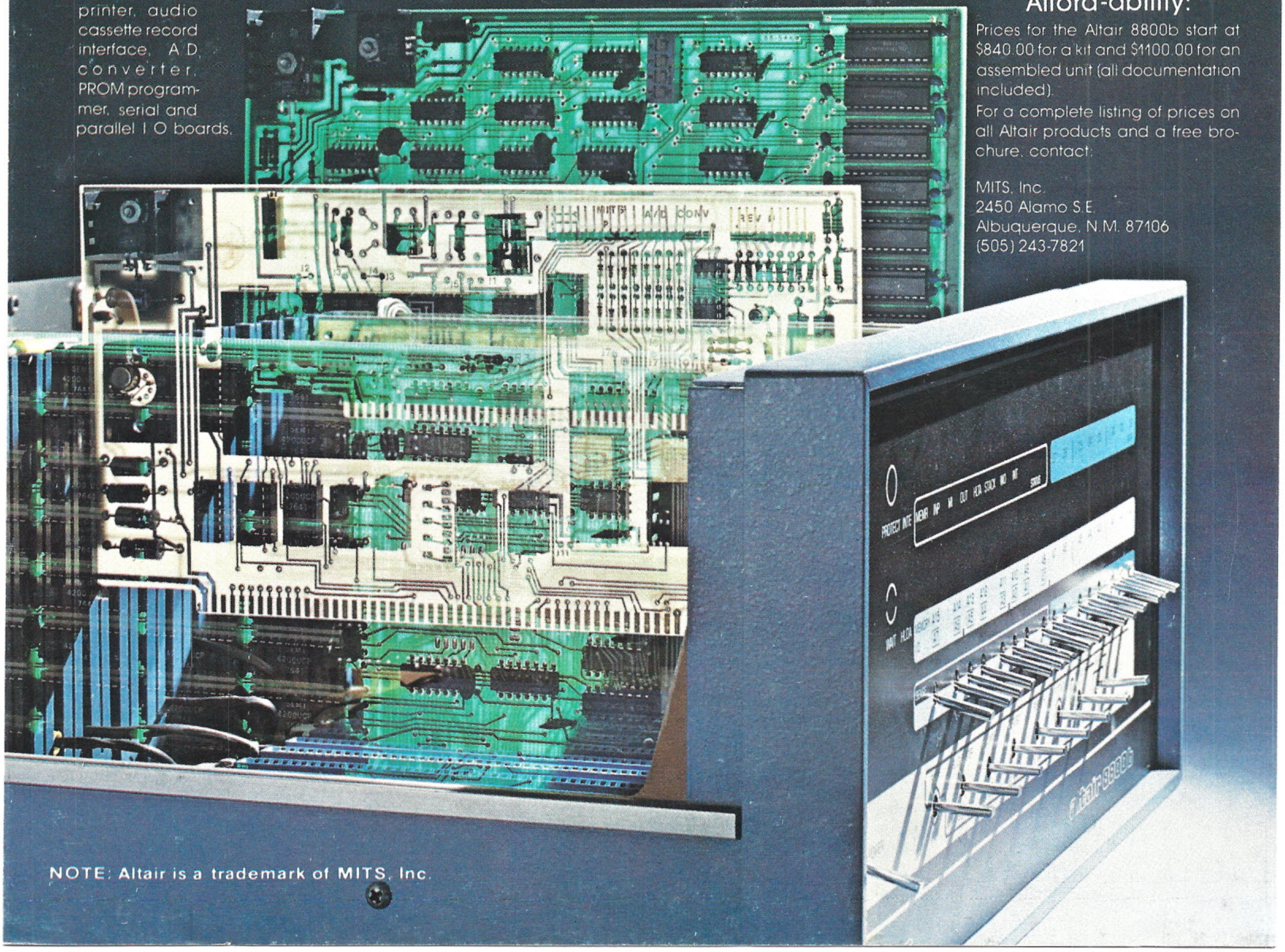
Package II, an assembly language development system for the Altair 8800b, includes system monitor, text editor, assembler and debug.

Afford-ability:

Prices for the Altair 8800b start at $840.00 for a kit and $1100.00 for an assembled unit (all documentation included).

For a complete listing of prices on all Altair products and a free brochure, contact:

MITS, Inc.
2450 Alamo S.E.
Albuquerque, N.M. 87106
(505) 243-7821



NOTE: Altair is a trademark of MITS, Inc.